AD657757

On the Structure of Sequential Machine Realizations

By
CHARLES A. HARLOW
CLARENCE L. COATES

July 19, 1967

Technical Report No. 27

N67-39517
(ACCESSION NUMBER)
81
(PAGES)
AD-657757
(NASA CR OR TMX OR AD NUMBER)
FACILITY FORM 602
CR-89531
3
(THRU)
(CODE)
08
(CATEGORY)

LABORATORIES FOR ELECTRONICS AND
RELATED SCIENCE RESEARCH
College of Engineering
THE UNIVERSITY OF TEXAS
AUSTIN, TEXAS 78712

Acquisitioned Document
SQT

This document has been approved for public release and sale; its distribution is unlimited.

The University of Texas' Laboratories for Electronics and Related Science Research are interdisciplinary laboratories in which graduate faculty members and graduate candidates from numerous academic disciplines conduct research.

Research conducted for this technical report was supported in part by the Department of Defense's JOINT SERVICES ELECTRONICS PROGRAM (U. S. Army, U. S. Navy, and the U. S. Air Force) through the Research Grant AF-AFOSR-766-67. This program is monitored by the Department of Defense's JSEP Technical Advisory Committee consisting of representatives from the U. S. Army Electronics Command, U. S. Army Research Office, Office of Naval Research, and the U. S. Air Force Office of Scientific Research.

Additional support of specific projects by other Federal Agencies, Foundations, and The University of Texas is acknowledged in footnotes to the appropriate sections.

Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

Requests for additional copies by agencies of the Department of Defense, their contractors and other government agencies should be directed to:

Defense Documentation Center (DDC)
Cameron Station
Alexandria, Virginia 22314

Department of Defense contractors must be established for DDC services to have their "need to know" certified by the cognizant military agency of their project or contract.

THE UNIVERSITY OF TEXAS

LABORATORIES FOR ELECTRONICS AND RELATED SCIENCE RESEARCH

ON THE STRUCTURE OF SEQUENTIAL MACHINE REALIZATIONS*

Technical Report No. 27

by

Charles A. Harlow**
Clarence L. Coates***

July 19, 1967

* Research sponsored by the National Science Foundation under Grants GP-2724 and GK-1146X, National Aeronautics and Space Administration under Grant NGR-44-012-449 and by the Joint Services Electronics Program under Grant AF-AFOSR-766-66.

** Graduate Student, Candidate for Ph.D. Degree in Electrical Engineering.

*** Professor of Electrical Engineering, The University of Texas.

ABSTRACT

In most studies of the structure of sequential machines there has been a tacit assumption that the machine was to be realized with unit delay memory elements. In this report we consider the sequential machines that are realized with either trigger or set-reset flip-flop memory elements.

It is shown that the relation called a partition pair which predicts the dependence of the input functions to unit delay memory elements does not predict the dependence of the input functions to trigger or set-reset flip-flop memory elements. In this paper we define relations called t-pairs and r-pairs which characterize the dependence of the input functions to trigger and set-reset flip-flop memory elements respectively. It is found that these relations do not have all the algebraic properties that partition pairs possess.

Feedback in sequential machines that are realized with trigger or set-reset flip-flop memory elements is also studied. A method is given for determining when a machine can be realized with either trigger or set-reset flip-flop memory elements using function f for feedback. It is shown that if a sequential machine can be realized with unit delay memory elements using a function f for feedback then it can be realized with set-reset flip-flops using f for feedback. It is also shown that for

completely specified machines that if a machine can be realized without feedback using unit delay memory elements then it cannot be realized without feedback using trigger flip-flop memory elements. The converse statement is also true.

LIST OF DEFINITIONS

TABLE OF CONTENTS

TABLE OF CONTENTS - (Continued)

CHAPTER I

Introductory Concepts

This report is concerned with the structure of synchronous sequential machines that are realized with trigger or set-reset flip-flop memory elements. Hartmanis and Stearns have considered this problem for delay type memory elements. A familiarity with their results which are in the references would be useful in understanding this paper. We shall now state some well known preliminary concepts.

Definition 1. A sequential machine is a five tuple $M = (\{s\}, \{x\}, \{0\}, \delta, \lambda)$. Where

1. $\{s\}$ is a finite set called the states of M.
2. $\{x\}$ is a finite set called the inputs to M.
3. $\{0\}$ is a finite set called the outputs of M.
4. δ is a function with the domain of δ a subset of $\{s\} \times \{x\}$ and range a subset of $\{s\}$. That is $\delta:\{s\} \times \{x\} \rightarrow \{s\}$.
5. λ is a function with domain a subset of $\{s\} \times \{x\}$ and range $\{0\}$. Thus $\lambda:\{s\} \times \{x\} \rightarrow \{0\}$.

For our purposes λ and $\{0\}$ are not important and we suppose the inputs to be n-tuples of $\{0,1\}$. Frequently we discuss partitions on the states of a machine M. These partitions will be denoted by Greek letters. A definition and an example of this concept follows.

Definition 2. A partition ρ on a set $\{s\}$ is a collection of subsets of $\{s\}$ such that

1. $\bigcup_{A \in \rho} A = \{s\}$

2. If A and B are in ρ, then $A \cap B = \varphi$.

Example. If $\{s\} = \{1,2,3,4,5\}$ then a partition ρ is given by $\rho = \{\{1,2\},\{3,4\},\{5\}\}$. It is more convenient, however, to use the notation $\rho = (\overline{1,2};\overline{3,4};\overline{5})$. The subsets of ρ are often called blocks of ρ. For example $\overline{1,2}$ is a block of ρ. When we discuss partitions we frequently need to discuss their blocks. If ρ is a partition on a set $\{s\}$ and if $a \in \{s\}$, then $\rho[a]$ will denote the block of ρ which contains a. In the above example $\rho[3] = \overline{3,4}$.

A trigger flip-flop is the two state sequential machine specified in Figure 1. A particular input to a trigger flip-flop will be denoted by T.

s	T	$\delta(s,T)$	$\lambda(s,T)$
1	0	1	0
1	1	2	0
2	0	2	1
2	1	1	1

$\{s\} = \{1,2\}$ states
$\{x\} = \{0,1\}$ inputs
$\{0\} = \{0,1\}$ output

Figure 1

If we are discussing more than one trigger flip-flop, we will index them with integers and refer to the i^{th} flip-flop with input T_i where i is an integer.

A set-reset flip-flop is the two state machine specified in Figure 2. A set-reset flip-flop has 2 inputs. A particular input will be denoted by (S,R) where S is called the set input and R is called the reset input. Again if we are discussing more than one set-reset flip-flop we shall index them with integers and refer to the i^{th} flip-flop with inputs R_i and S_i where i is an integer.

s	S	R	$\delta(s,S,R)$	$\lambda(s,S,R)$
1	0	0	1	0
1	0	1	1	0
1	1	0	2	0
2	0	0	2	1
2	0	1	1	1
2	1	0	2	1

$\{s\} = \{1,2\}$

$\{x\} = \{0,1\} \times \{0,1\}$

$\{0\} = \{0,1\}$

Figure 2

The next definition is that of a weak partition pair which is discussed in Reference 3. Our notation for a partition pair will be somewhat different than that of Hartmanis and Stearns.

Definition 3. Let τ and ρ be state partitions on machine $M = (\{s\},\{x\},\{0\},\delta,\lambda)$. Then τ "p" ρ iff $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ for every two states a and b such that $\tau[a] = \tau[b]$ and for every input x such that $\delta(a,x)$ and $\delta(b,x)$ are defined.

The following results are some properties of weak partition pairs which we shall merely list since they are proved in Reference 3. But first we must define what we mean by one partition being greater than another and we must also define how we multiply and add partitions.

Definition 4. Let τ, ρ be partitions on $\{s\}$. Then $\tau \geq \rho$ iff $\tau[a] \geq \rho[a]$ for every $s \in \{s\}$. Recall that $\tau[a]$ and $\rho[a]$ are sets.

For example if $\{s\} = \{1,2,3,4,5\}$, $\tau = (\overline{1,2;3,4,5})$ and $\rho = (\overline{1,2;3,4;5})$ then $\tau \geq \rho$.

Definition 5. Let τ and ρ be partitions on $\{s\}$. Then $\tau + \rho = \gamma$ where γ is the smallest partition (equivalence relation) which contains both τ and ρ. This is characterized by the property that if $a, b \in \{s\}$ then $\gamma[a] = \gamma[b]$ if and only if there exists $a_1, \ldots, a_k$ in $\{s\}$ such that $a = a_1$, $a_k = b$ and for every i such that $1 \leq i \leq k-1$ either $\tau[a_i] = \tau[a_{i+1}]$ or $\rho[a_i] = \rho[a_{i+1}]$.

Example. Let $\{s\} = \{1,2,3,4,5\}$, $\rho = (\overline{1,2;3,4;5})$ and $\tau = (\overline{1;2,3;4;5})$. Then $\tau + \rho = (\overline{1,2,3,4;5})$.

Definition 6. Let τ and ρ be partitions on $\{s\}$. Then $\tau \cdot \rho = \gamma$ where γ is a partition on $\{s\}$ such that for every $a, b \in \{s\}$ $\gamma[a] = \gamma[b]$ iff $\tau[a] = \tau[b]$ and $\rho[a] = \rho[b]$.

Result 1 (Lemma).

Let $M = (\{s\}, \{x\}, \{0\}, \delta, \lambda)$ be a machine.

1. If τ"p"ρ and if $\gamma \geq \rho$, then τ"p"γ.
2. If τ"p"ρ and if $\alpha \leq \tau$, then α"p"ρ.
3. If M is completely specified; that is, if the domain of δ is $\{s\} \times \{x\}$, then τ_1"p"ρ_1 and τ_2"p"ρ_2 implies that $(\tau_1 + \tau_2)$"p"$(\rho_1 + \rho_2)$.
4. If τ_1"p"ρ_1 and τ_2"p"ρ_2 then $\tau_1 \cdot \tau_2$"p"$\rho_1 \cdot \rho_2$.

In order to realize a machine M it is necessary to code the states of M into n tuples of $\{0,1\}$. The coding function will be called h and $h:\{s\} \rightarrow \{0,1\}^n$ is a 1-1 function. The i^{th} projection of h will be called h_i that is for every state $h_i(s) = y_i$ where $h(s) = (y_1, \ldots, y_i, \ldots, y_n)$ and y_i is in $\{0,1\}$. It should be noted that our concept of a realization and that of Reference 3 are not the same in that we do not expand the machine.

We are often interested in subspaces of $\{0,1\}^n$. To be specific, let $n = 5$. $(y_1, \ldots, y_n) = (0,1,0,1,1)$ is in $\{0,1\}^5$. We want a general way to refer to specific coordinates of $(y_1, \ldots, y_n)$, say coordinates 3 and 5 where $(y_3, y_5) = (0,1)$ an element of $\{0,1\}^2$. We will use the following formalism to do this. If $\Lambda \leq [1, \ldots, n]$, we let $\{0,1\}^\Lambda$ be $\{0,1\}^j$ where j is the number of elements in Λ. If $y = (y_1, \ldots, y_n) \in \{0,1\}^n$, we

denote by y_Λ the j tuple $(y_{i_1}, \ldots, y_{i_j}) \in \{0,1\}^\Lambda$ where $i_1 < i_2 < \cdots < i_j$ and $i_1, i_2, \ldots, i_j$ are all in Λ. In the above example $\Lambda = \{3,5\}$, $\{0,1\}^\Lambda = \{0,1\}^2$ and when $y = (y_1, \ldots, y_n) = (0,1,0,1,1)$ then $y_\Lambda = (0,1) \in \{0,1\}^\Lambda$.

If we are given a coding function $h:\{s\} \to \{0,1\}^n$ for a machine M we can associate the following partitions with it. For every $i \in \{1, \ldots, n\}$ we define the partition ρ_i by $\rho_i[a] = \rho_i[b]$ iff $h_i(a) = h_i(b)$. We call ρ_i the partition associated with h_i. Conversely given a two block partition ρ_i on a machine M we can define a function h_i on $\{s\}$ such that $h_i(a) = 1$ if a is in block 1 of ρ_i and $h_i(a) = 0$ if a is in block 2 of ρ_i. This h_i will be called the function associated with ρ_i. If there are n such ρ_i then $h(a) = (h_1(a), \ldots, h_n(a))$ is 1-1 if $\prod_1^n \rho_i = \emptyset$ the zero partition. Often we are given $\Lambda \leq \{1, \ldots, n\}$ and we want to discuss $h_\Lambda(a)$ for $a \in \{s\}$. It should be noted that if $\tau = \prod_\Lambda \rho_i$ then $\tau[a] = \tau[b]$ implies that $h_\Lambda(b)$.

If we code a machine by h into $\{0,1\}^n$ then h need not be onto. We denote by Y_i a function such that $Y_i:\{0,1\}^n \times \{x\} \to \{0,1\}$ and $Y_i(h(a), x) = h_i(\delta(a,x))$ for every $a \in \{s\}$ and $x \in \{x\}$. Thus Y_i is an extension of the i^{th} next state function to all of $\{0,1\}^n$.

A problem in many of our results is filling in the "don't care" terms properly. In the results pertaining to unit delay realizations this is fairly easy. But when one considers flip-flop realizations the

situation is more complicated. This is the reason for much of the complexity in some of our proofs relating to reduced dependence. With this in mind we state the following Theorem.

Result 2 (Theorem).

Let M be a machine coded by h into $\{0,1\}^n$. Let Λ and Λ' be subsets of $\{1,\ldots,n\}$. If $\tau = \prod_{\Lambda} \rho_i$ and $\rho = \prod_{\Lambda'} \rho_i$, then τ "p" ρ iff for every $g \epsilon \Lambda'$ there exists Y_g such that $Y_g(y,x) = F_g(y_\Lambda, x)$ for every input x and for every $y \epsilon \{0,1\}^n$.

Proof.

For every $y \epsilon \{0,1\}^n$ such that $y = h(a)$ for some $a \epsilon \{s\}$ and such that $\delta(a,x)$ is specified let $Y_g(y,x) = h_g[\delta(a,x)]$ for every $g \epsilon \Lambda'$. Suppose τ "p" ρ. Show that if we define $F_g(h_\Lambda(a),x) = h_g[\delta(a,x)]$ then F_g is well defined on $\{h_\Lambda(a) \mid a \epsilon \{s\}\}$. If there exists $a,b \epsilon \{s\}$ such that $h_\Lambda(a) = h_\Lambda(b)$ and $\delta(a,x)$ and $\delta(b,x)$ are specified, then $\tau[a] = \tau[b]$. Since τ "p" ρ this implies $\rho[\delta(a,x)] = \rho[\delta(b,x)]$. Therefore $\rho_g[\delta(a,x)] = \rho_g[\delta(b,x)]$ because $\rho_g \geq \rho$. Thus $h_g[\delta(a,x)] = h_g[\delta(b,x)]$ and F_g is well defined on the set of all $\{h_\Lambda(a) \mid a \epsilon \{s\}\}$. Extend F_g to all of $\{0,1\}^\Lambda$ in any manner. Recall that $F_g(y_\Lambda, x)$ is arbitrary if there is no $a \epsilon \{s\}$ and $x \epsilon \{x\}$ with $\delta(a,x)$ specified and $y_\Lambda = h_\Lambda(a)$. For every $y \epsilon \{0,1\}^n$ let $Y_g(y,x) = F_g(y_\Lambda, x)$.

Consider the converse. Let $a,b \epsilon \{s\}$ and $x \epsilon \{x\}$ such that $\delta(a,x)$ and $\delta(b,x)$ are specified and $\tau[a] = \tau[b]$. $\tau[a] = \tau[b]$ implies that

$h_\Lambda(a) = h_\Lambda(b)$. Therefore $Y_g(h(a),x) = h_g(\delta(a,x)) = F_g(h_\Lambda(a),x) = F_g(h_\Lambda(b),x) = h_g[\delta(b,x)]$ for every $g\epsilon\Lambda$. This in turn implies that $\rho_g[\delta(a,x)] = \rho_g[\delta(b,x)]$ for every $g\epsilon\Lambda$. Hence $\rho[\delta(a,x)] = \rho[\delta(b,x)]$. ‖

This concludes the preliminaries. We will now consider sequential machines that are realized with flip-flop memory elements. In the next chapter we consider first machines that are realized with trigger flip-flop memory elements and later consider machines that are realized with set-reset flip-flop memory elements.

CHAPTER II

Realizations with Flip-Flop Memory Elements

Trigger Flip-Flop Realizations

The following example demonstrates the difference between trigger flip-flop realizations and unit delay realizations. Machine A in Figure 3 has partition pairs $(\overline{1,2};\overline{3,4})$"p"$(\overline{1,3};\overline{2,4})$ and $(\overline{1,3};\overline{2,4})$"p"$(\overline{1,2};\overline{3,4})$ where $(\overline{1,2};\overline{3,4}) \cdot (\overline{1,3};\overline{2,4}) = \emptyset$. If we let ρ_1 be $(\overline{1,3};\overline{2,4})$ and associate h_1 with ρ_1 and if we let ρ_2 be $(\overline{1,2};\overline{3,4})$ and associate h_2 with ρ_2 then from Result 2 it follows that $Y_1(y_1, y_2, x) = F_1(y_2, x)$ and $Y_2(y_1, y_2, x) = F_2(y_1, x)$ where $y_1, y_2 \in \{0, 1\}$ amd $x \in \{x\}$. In particular $Y_1 = \bar{x}y_2 + x\bar{y}_2$ and $Y_2 = \bar{x}y_1 + x$. But when we compute the trigger functions we get $T_1(y_1, y_2, x) = \bar{x}\bar{y}_1y_2 + \bar{x}y_1\bar{y}_2 + x\bar{y}_1\bar{y}_2 + xy_1y_2$ and $T_2(y_1, y_2, x) = \bar{x}\bar{y}_1y_2 + y_1\bar{y}_2 + x\bar{y}_2$ where the operations $\cdot$ and $+$ are Boolean.

Inputs

States	0	1
1	1	4
2	3	4
3	2	3
4	4	3

δ

$(\overline{1,2};\overline{3,4})$"p"$(\overline{1,3};\overline{2,4})$
$(\overline{1,3};\overline{2,4})$"p"$(\overline{1,2};\overline{3,4})$

Figure 3. Machine A

We see from this example that the functional dependence is not in general the same for trigger flip-flop realizations as it is for delay realizations. Hence partition pairs do not characterize trigger flip-flop realizations. The next definition gives a relation which does characterize trigger flip-flop realizations.

Definition 7. Let $M = (\{s\},\{x\},\{0\},\delta,\lambda)$ be a sequential machine. The state partitions τ and ρ are in relation τ "t" ρ iff

1. ρ is a two block partition
2. $\tau \cdot \rho$ "p" ρ
3. For every two states a,b such that $\tau[a] = \tau[b]$ and $\rho[a] \neq \rho[b]$ and for every input x, then $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ when $\delta(a,x)$ and $\delta(b,x)$ are specified.

Note that 3 merely says that the next states are in different blocks of ρ. For an example of this definition consider Machine B in Figure 4. In this machine $(\overline{1,2};\overline{3,4,5})$ "t" $(\overline{1,2,3,5};\overline{4})$. Note that $(\overline{1,2,3,5};\overline{4})$ is a two block partition and that $(\overline{1,2};\overline{3,5};\overline{4})$ "p" $(\overline{1,2,3,5};\overline{4})$ where $\tau \cdot \rho = (\overline{1,2};\overline{3,4,5}) \cdot (\overline{1,2,3,5};\overline{4}) = (\overline{1,2};\overline{3,5};\overline{4})$. Also for every input x $\delta(3,x)$ and $\delta(4,x)$ as well as $\delta(5,x)$ and $\delta(4,x)$ are in opposite blocks of $(\overline{1,2,3,5};\overline{4})$.

		Inputs	
		0	1
	1	3	3
States	2	5	1
	3	2	4
	4	4	2
	5	1	4

$\overline{(1,2;\overline{3,4,5})}$"t"$\overline{(1,2,3,5;\overline{4})}$

Figure 4. Machine B

It should be observed that if $\rho \geq \tau$ then τ"p"ρ and τ"t"ρ are equivalent when ρ is a two block partition. Also if ρ is a two block partition and $\tau = \rho$ then τ"t"ρ implies ρ"p"ρ which implies that ρ has the substitution property.

Before we proceed to the principal results we need the following lemma.

Result 3 (Lemma).

Let machine M be realized with the g^{th} memory element a trigger flip-flop and let Λ be a subset of $\{1,\ldots,n\}$ which does not contain g. Also suppose $Y_g(y,x) = y_g\, M(y_\Lambda, x) + \bar{y}_g\, N(y_\Lambda, x)$ for every input x and for every n tuple y in $\{0,1\}^n$. Then $T_g(y,x) = G_g(y_\Lambda, x)$ iff $M = \bar{N}$.

Proof.

i) Suppose $T_g(y,x) = G_g(y_\Lambda, x)$. In general $T_g(y,x) = y_g \bar{Y}_g(y,x) + \bar{y}_g Y_g(y,x)$. Substituting the equation given for Y_g and simplifying gives

$T_g(y,x) = y_g \bar{M}(y_\Lambda, x) + \bar{y}_g N(y_\Lambda, x)$. We claim $M = \bar{N}$. Suppose there exists y_Λ and x such that $M(y_\Lambda, x) \neq \bar{N}(y_\Lambda, x)$. Consider the n tuple y' where $y'_i = y_i$ for every i in Λ, $y'_g = 0$ while the remaining y_i are arbitrary. Note that g is not in Λ. Consider the n tuple $y^\#$ where $y^\#_i = y_i$ for every i in Λ, $y^\#_g = 1$ while the remaining $y^\#_i$ are arbitrary. Then $T_g(y', x) = N(y_\Lambda, x)$ and $T_g(y^\#, x) = \bar{M}(y_\Lambda, x)$. But from the hypothesis $T_g(y', x) = T_g(y^\#, x)$. Therefore $N(y_\Lambda, x) = \bar{M}(y_\Lambda, x)$ which is a contradiction.

ii) Suppose $M = \bar{N}$. As before $T_g(y,x) = Y_g \bar{M}(y_\Lambda, x) + \bar{y}_g N(y_\Lambda, x)$. Since $M = \bar{N}$ this implies that $T_g(y,x) = N(y_\Lambda, x)$ for every n tuple y in $\{0,1\}^n$ and for every input x. ‖

Now we give the theorems for reduced dependence for trigger flip-flop input functions. The next two theorems give necessary and sufficient conditions that the input function to a trigger flip-flop be a function of a subset of the state variables.

Result 4 (Theorem).

If a sequential machine M coded into $\{0,1\}^n$ by h has a realization with the g^{th} memory element a trigger flip-flop such that $T_g(y,x) = G_g(y_\Lambda, x)$ where y is any element of $\{0,1\}^n$, x is an input, and Λ is a subset of $\{1,\ldots,n\}$ then the state partitions ρ and τ are such that τ "t" ρ where $\rho = \rho_g$ and $\tau = \prod_\Lambda \rho_i$.

Proof.

i) Suppose $g \in \Lambda$. Then $\tau \leq \rho$ and since for a trigger flip-flop $Y_g(y,x) = T_g(y,x)\ \bar{y}_g + \bar{T}_g(y,x)\ y_g$ we get that $Y_g(y,x) = G_y(y_\Lambda, x)\ \bar{y}_g + \bar{G}_g(y_\Lambda, x)\ y_g$ from the hypothesis. Therefore $Y_g(y,x) = F_g(y_\Lambda, x)$ since $g \in \Lambda$. Thus from Result 2 we get that τ "p" ρ. Since $\rho \geq \tau$ this means τ "t" ρ.

ii) Suppose $g \notin \Lambda$. From the hypothesis and Result 3 $Y_g(y,x) = y_g\ M(y_\Lambda, x) + \bar{y}_g\ N(y_\Lambda, x)$ for every $y \in \{0,1\}^n$ and for every input x and also $M = \overline{N}$. From Result 2 this implies that $\tau \cdot \rho$ "p" ρ since $Y_g(y,x) = F_g(y_\Lambda, y_g, x)$.

Let a,b be states such that $\tau[a] = \tau[b]$ and $\rho[a] \neq \rho[b]$ and let x be an input. Show $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ if $\delta(a,x)$ and $\delta(b,x)$ are specified. $\tau[a] = \tau[b]$ implies that $h_\Lambda(a) = h_\Lambda(b)$ from the hypothesis and the definition of ρ. Also $\rho[a] \neq \rho[b]$ implies that $h_g(a) \neq h_g(b)$ from the definition of $\rho = \rho_g$. But $Y_g(h(a),x) = h_g(a)\ M(h_\Lambda(a),x) + \bar{h}_g(a)\ N(h_\Lambda(a),x)$ and $Y_g(h(b),x) = h_g(b)\ M(h_\Lambda(b),x) + \bar{h}_g(b)\ N(h_\Lambda(b),x)$. Since $h_g(a) \neq h_g(b)$ assume with no loss of generality that $h_g(a) = 1$ and $h_g(b) = 0$. Then $Y_g(h(a),x) = M(h_\Lambda(a),x)$ and $Y_g(h(b),x) = N(h_\Lambda(b),x)$. Since $h_i(a) = h_i(b)$ for every $i \in \Lambda$ and $M = \overline{N}$ this implies that $Y_g(h(a),x) \neq Y_g(h(b),x)$ which implies that $h_g[\delta(a,x)] \neq h_g[\delta(b,x)]$ from the definition of ρ_g. ‖

A corollary to this theorem relates delay realizations to trigger flip-flop realizations.

Result 5 (Corollary).

Let M be a machine coded by h into $\{0,1\}^n$ and let Λ, Λ' be subsets of $\{1,\ldots,n\}$ with $\Lambda' \leq \Lambda$. If machine M is realized with trigger flip-flop memory elements such that $T_i(y,x) = G_i(y_\Lambda, x)$ for every i in Λ' then the partitions $\tau = \prod_{\Lambda} \rho_i$ and $\rho = \prod_{\Lambda'} \rho_i$ are such that τ"p"ρ.

Proof.

From Result 4 τ"t"ρ_i for every i in Λ. Since $i \in \Lambda'$ implies $i \in \Lambda$ we have that $\tau \leq \rho_i$. Therefore τ"t"ρ_i implies τ"p"ρ_i for every i in Λ'. Thus τ"p" $\prod_{\Lambda'} \rho_i$ since the multiplication of partition pairs is a partition pair from Result 1. Thus τ"p"ρ.

We now give the necessary conditions for reduced dependence for trigger flip-flop realizations.

Result 6 (Theorem).

If machine M is coded by h into $\{0,1\}^n$ such that τ"t"ρ where $\rho = \rho_g$ with $g \in \{1,\ldots,n\}$ and $\tau = \prod_{\Lambda} \rho_i$ with $\Lambda \leq \{1,\ldots,n\}$ then $T_g(y,x) = G_g(y_\Lambda, x)$ for every y in $\{0,1\}^n$ and for every input x.

Proof.

Since τ"t"ρ we know that $\tau \cdot \rho$"p"ρ. From Result 2 this implies that $Y_g(y,x) = F_g(y_\Lambda, y_g, x)$ for every y in $\{0,1\}^n$ and for every input x.

There are two cases to consider.

i) Suppose $g \in \Lambda$. Then $Y_g(y,x) = F_g(y_\Lambda, x)$. Recall that $T_g(y,x) = y_g \bar{Y}_g(y,x) + \bar{y}_g Y_g(y,x) = y_g \bar{F}_g(y_\Lambda, x) + \bar{y}_g F_g(y_\Lambda, x)$. If we let $G_g(y_\Lambda, x) = y_g \bar{F}_g(y_\Lambda, x) + \bar{y}_g F_g(y_\Lambda, x)$, then we get the result.

ii) Suppose $g \notin \Lambda$. Recall that we have certain freedoms on F_g. Namely for every $a \in \{s\}$ and input x such that $\delta(a,x)$ is specified $F_g(h_\Lambda(a), h_g(a), x) = h_g[\delta(a,x)]$. Otherwise we may specify $F_g(y_\Lambda, y_g, x)$ in any manner. We specify F_g on all the (y_Λ, y_g) such that $(y_\Lambda, y_g) \neq (h_\Lambda(a), h_g(a))$ for any $a \in \{s\}$ with $\delta(a,x)$ specified as follows. If $y_\Lambda = h_\Lambda(a)$ for some $a \in \{s\}$ but $y_g \neq h_g(a)$, then define $F_g(y_\Lambda, y_g, x) = \bar{F}_g(h_\Lambda(a), h_g(a), x)$. For the (y_Λ, y_g) that remain we define F_g with the constraint that $F_g(y_\Lambda, y_g, x) = \bar{F}_g(y_\Lambda, \bar{y}_g, x)$ which can clearly be done. Show $F_g(y_\Lambda, y_g, x) = \bar{F}_g(y_\Lambda, y_g, x)$ for all the (y_Λ, y, x). The only case that must be considered is when there exists $a, b \in \{s\}$ and x such that $h_i(a) = h_i(b)$ for every i in Λ and $h_g(a) \neq h_g(b)$ while $\delta(a,x)$ and $\delta(b,x)$ are specified. But $h_i(a) = h_i(b)$ for every i in Λ implies $\tau[a] = \tau[b]$ and $h_g(a) \neq h_g(b)$ implies $\rho[a] \neq \rho[b]$. From the definition of τ "t" ρ this implies that $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ or $h_g[\delta(a,x)] \neq h_g[\delta(b,x)]$ which is equivalent to $Y_g(h(a),x) \neq Y_g(h(b),x)$. Therefore $F_g(h_\Lambda(a), h_g(a), x) \neq F_g(h_\Lambda(b), h_g(b), x)$. Thus $F_g(y_\Lambda, y_g, x) \neq F_g(y_\Lambda, \bar{y}_g, x)$ for all (y_Λ, y_g, x).

Since $Y_g(y,x) = F_g(y_\Lambda, y_g, x)$ we deduce that $Y_g(y,x) = y_g M(y_\Lambda, x) + \bar{y}_g N(y_\Lambda, x)$. Show $M = \bar{N}$. Consider a particular (y_Λ, x),

then $M(y_\Lambda, x) = F_g(y_\Lambda, y_g = 1, x)$ and $N(y_\Lambda, x) = F_g(y_\Lambda, y_g = 0, x)$. But $F_g(y_\Lambda, 1, x) = \bar{F}_g(y_\Lambda, 0, x)$ thus $N(y_\Lambda, x) = \bar{M}(y_\Lambda, x)$. From Result 3 this says that $T_g(y, x) = G_g(y_\Lambda, x)$. ‖

It should be noted in the proof that when there are "don't care" terms they must be filled in properly. Recall that these terms arise from an incompletely specified machine or they arise when all the n tuples of $\{0,1\}^n$ do not represent a state as happens in 5 state machines.

Result 7 (Corollary).

Let machine M have state partitions τ and ρ such that τ "p" ρ and $\rho \geq \tau$. If M is coded by h into $\{0,1\}^n$ such that $\tau = \prod_\Lambda \rho_i$ where $\Lambda \leq \{1, \ldots, n\}$ and $\rho \leq \rho_g$ where $g \in \{1, \ldots, n\}$ then $T_g(y, x) = G_g(y_\Lambda, x)$ for every y in $\{0,1\}^n$ and for every input x.

Proof.

If $\rho_g \geq \rho$ then τ "p" ρ implies τ "p" ρ_g from Result 1. $\rho_g \geq \rho \geq \tau$ means that τ "p" ρ_g is equivalent to τ "t" ρ_g. Therefore from Result 6 $T_g(y, x) = G_g(y_\Lambda, x)$. ‖

For an example of Result 7 consider machine C in Figure 5. If we code machine C with h such that h_1 is constant on the blocks of ρ_1, h_2 is constant on the blocks of ρ_2 and h_3 is constant on the blocks of ρ_3, then from Result 6 $T_1(y_1, y_2, y_3, x) = G_1(y_2, x)$ since ρ_2 "t" ρ_1. For one possible assignment $T_1(y_1, y_2, y_3, x) = y_2\bar{x}$.

States	Inputs 0	Inputs 1
1	2	3
2	2	4
3	5	1
4	5	4
5	3	5

$\rho_1 = (\overline{1,2,3,4};\overline{5})$

$\rho_2 = (\overline{1,2};\overline{3,4,5})$

$\rho_3 = (\overline{1,3,5};\overline{2,4})$

ρ_2"t"ρ_1 or $(\overline{1,2};\overline{3,4,5})$"5"$(\overline{1,2,3,4};\overline{5})$

Figure 5. Machine C

It is obvious that the relation "t" does not have all the algebraic properties of the relation "p" since for all partitions such that τ"t"ρ we limit ρ to a two block partition. Therefore we cannot expect statements like the multiplication of t-pairs is again a t-pair. Nevertheless the relation "t" does have some properties which we now investigate.

Result 8.

If τ"t"ρ and $\gamma < \tau$ then γ"t"ρ.

Proof.

i) Show $\gamma \cdot \rho$"p"ρ. $\gamma \leq \tau$ implies $\gamma \cdot \rho \leq \tau \cdot \rho$ hence from Result 1 and the fact that $\tau \cdot \rho$"p"ρ we deduce that $\gamma \cdot \rho$"p"ρ.

ii) Let $a,b \in \{s\}$ such that $\gamma[a] = \gamma[b]$ and $\rho[a] \neq \rho[b]$. Then since $\tau \geq \gamma$ we have that $\tau[a] = \tau[b]$. Because τ"t"ρ this implies that $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ if $\delta(a,x)$, $\delta(b,x)$ are specified. ‖

Result 9.

If τ"t"ρ and γ"t"ρ then $\tau \cdot \gamma$"t"ρ.

Proof.

$\tau \cdot \gamma \leq \tau$ implies $\tau \cdot \gamma$"t"ρ from Result 8.

Result 10.

Let M be completely specified. If τ"t"ρ and γ"t"ρ then $(\tau + \gamma)$"t"ρ.

Proof.

i) First consider a sequence of states $a_1, \ldots, a_j$ such that for every i in $\{1, \ldots, j-1\}$ $\tau[a_i] = \tau[a_{i+1}]$ or $\gamma[a_i] = \gamma[a_{i+1}]$. We show by induction that if $\rho[a_1] = \rho[a_j]$ then $\rho[\delta(a_1, x)] = \rho[\delta(a_j, x)]$ while if $\rho[a_1] \neq \rho[a_j]$ that $\rho[\delta(a_1, x)] \neq \rho[\delta(a_j, x)]$.

Suppose $j = 2$. Then $\tau[a_1] = \tau[a_2]$ or $\gamma[a_1] = \gamma[a_2]$. If $\rho[a_1] = \rho[a_2]$ this implies $\rho[\delta(a_1, x)] = \rho[\delta(a_2, x)]$ since $\tau \cdot \rho$"p"ρ and $\gamma \cdot \rho$"p"ρ. If $\rho[a_1] \neq \rho[a_2]$ then we deduce that $\rho[\delta(a_1, x)] \neq \rho[\delta(a_2, x)]$ since τ"t"ρ and γ"t"ρ.

Assume the hypothesis is true for every integer j in $\{1, \ldots, k\}$ with $k \geq 2$. Let $a_1, \ldots, a_k, a_{k+1}$ be a sequence of states such that $\tau[a_i] = \tau[a_{i+1}]$ or $\gamma[a_i] = \gamma[a_{i+1}]$ for all i in $\{1, \ldots, k\}$.

Suppose $\rho[a_1] = \rho[a_{k+1}]$. If $\rho[a_k] = \rho[a_{k+1}]$ then $\rho[a_1] = \rho[a_k]$ which from the inductive hypothesis implies $\rho[\delta(a_1, x)] = \rho[\delta(a_k, x)] = \rho[\delta(a_{k+1}, x)]$. If $\rho[a_k] \neq \rho[a_{k+1}]$, then $\rho[a_1] \neq \rho[a_k]$

which implies $\rho[\delta(a_k, x)] \neq \rho[\delta(a_{k+1}, x)]$ and $\rho[\delta(a_1, x)] \neq \rho[\delta(a_k, x)]$ from the induction hypothesis. But since ρ is a two block partition this implies $\rho[\delta(a_1, x)] = \rho[\delta(a_{k+1}, x)]$.

Suppose $\rho[a_1] \neq \rho[a_{k+1}]$. If $\rho[a_k] = \rho[a_{k+1}]$ then $\rho[a_1] \neq \rho[a_k]$ which from the inductive hypothesis implies $\rho[\delta(a_1, x)] \neq \rho[\delta(a_k, x)] = \rho[\delta(a_{k+1}, x)]$ or $\rho[\delta(a_1, x)] \neq \rho[\delta(a_{k+1}, x)]$. If $\rho[a_k] \neq \rho[a_{k+1}]$ then $\rho[a_1] = \rho[a_k]$ which implies $\rho[\delta(a_1, x)] = \rho[\delta(a_k, x)] \neq \rho[\delta(a_{k+1}, x)]$ or $\rho(\delta(a_1, x)] \neq \rho[\delta(a_{k+1}, x)]$.

ii) Show $(\tau + \gamma)$"t"ρ. Let a,b be states such that $(\tau + \gamma)[a] = (\tau + \gamma)[b]$ then there exists a sequence of states $a_1, \ldots, a_j$ such that $a_1 = a$, $a_j = b$ and $\tau[a_i] = \tau[a_{i+1}]$ or $\gamma[a_i] = \gamma[a_{i+1}]$ for every i in $\{1, \ldots, j-1\}$. If $\rho[a] = \rho[b]$ then from i) $\rho[\delta(a, x)] = \rho[\delta(b, x)]$. Therefore $\rho \cdot (\tau + \gamma)$"p"$\rho$. If $\rho[a] \neq \rho[b]$ then from i) $\rho[\delta(a, x)] \neq \rho[\delta(b, x)]$. Therefore $(\tau + \gamma)$"t"ρ. ‖

For a given 2 block partition ρ on a completely specified machine Result 10 implies the existence of a largest partition τ such that τ"t"ρ.

We conclude this section with the following theorem. It gives the conditions which allow one to realize a machine with trigger flip-flops and get reduced dependence for more than one flip-flop.

Result 11 (Theorem).

Consider a machine M such that

i) There exists the following partitions in the t-relation $\tau_1 "t" \rho_1, \tau_2 "t" \rho_2, \ldots, \tau_n "t" \rho_n$.

ii) For every g in $\{1,\ldots,n\}$ there is a subset Λ_g of $\{1,\ldots,n\}$ such that

$$\tau_g \geq \prod_{\Lambda_g} \rho_j$$

iii) For every g in $\{1,\ldots,n\}$ M is coded by h into $\{0,1\}^n$ such that h_g is constant on ρ_g. In other words ρ_g is the partition associated with h_g.

Then for every g in $\{1,\ldots,n\}$ $T_g(y,x) = G_g(y_\Lambda, x)$.

Proof.

It follows from i), ii) and Result 8 that for every g in $\{1,\ldots,n\}$ $(\prod_{\Lambda_g} \rho_j)"t"\rho_g$. From iii) and Result 6 this implies that $T_g(y,x) = G_g(y_\Lambda, x)$ for every y in $\{0,1\}^n$ and for every input x. ‖

An example of Result 11 is given by machine C in Figure 6. If we code machines C by h into $\{0,1\}^3$ such that ρ_1, ρ_2, ρ_3 are the partitions associated with h then since

$\tau_1 \geq \rho_2$

$\tau_2 \geq \rho_1 \cdot \rho_2 \cdot \rho_3$

$\tau_3 \geq \rho_2 \cdot \rho_3$

we deduce from Result 12 that $T_1(y_1, y_2, y_3, x) = G_1(y_2, x)$, $T_2(y_1, y_2, y_3, x) = G_2(y_1, y_2, y_3, x)$, and $T_3(y_1, y_2, y_3, x) = G_3(y_2, y_3, x)$. For one such coding function h $T_1 = y_2\bar{x}$, $T_2 = \bar{y}_2 x + \bar{y}_1 y_2 \bar{y}_3 x$ and $T_3 = \bar{y}_2 \bar{y}_3 \bar{x} + y_2 y_3 \bar{x}$.

		Inputs 0	1
	1	2	3
	2	2	4
States	3	5	1
	4	5	4
	5	3	5

$\overline{(1,2;3,4,5)}$"t"$\overline{(1,2,3,4;5)} = \tau_1$"t"$\rho_1$

$\overline{(1,2,3;4;5)}$"t"$\overline{(1,2;3,4,5)} = \tau_2$"t"$\rho_2$

$\overline{(2,3,5;1,4)}$"t"$\overline{(1,3,5;2,4)} = \tau_3$"t"$\rho_3$

Figure 6. Machine C

Set-Reset Flip-Flop Realizations

In this section we consider sequential machines realized using set-reset flip-flops as memory elements. The organization and results of the section are similar to those of the previous section. As in the case for machines realized with trigger flip-flop memory elements partition pairs do not characterize machines realized with set-reset flip-flop memory elements. Definition 8 gives a relation which does characterize set-reset flip-flop realizations.

Definition 8. Let $M = (\{s\}, \{x\}, \{0\}, \delta, \lambda)$ be a sequential machine. The state partitions τ and ρ are in relation τ"r"ρ iff

1. ρ is a two block partition

2. $\tau \cdot \rho$"p"ρ

3. For every two states a and b such that $\tau[a] = \tau[b]$, $\rho[a] \neq \rho[b]$ and for every input x such that $\delta(a,x)$ and $\delta(b,x)$ are specified then either $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ or $\delta(a,x) \in \rho[a]$ and $\delta(b,x) \in \rho[b]$.

For an example of the preceding definition consider Machine D in Figure 7. In this machine $(\overline{1,2,3,4};\overline{5})$"r"$(\overline{1,2};\overline{3,4,5})$. Note $\rho = (\overline{1,2};\overline{3,4,5})$ is a two block partition and that $\tau \cdot \rho$"p"$\rho = (\overline{1,2};\overline{3,4};\overline{5})$ "p"$(\overline{1,2};\overline{3,4,5})$. Also note for example that $\rho[\delta(1,0)] = \rho[\delta(3,0)]$ and that $\delta(1,1) = 2$ is in $\rho[1] = \overline{1,2}$ and $\delta(3,1) = 4$ is in $\rho[3] = \overline{3,4,5}$.

		Inputs 0	1
	1	3	2
	2	4	2
States	3	5	4
	4	3	4
	5	1	5

τ"r"$\rho = (\overline{1,2,3,4};\overline{5})$"r"$(\overline{1,2};\overline{3,4,5})$

Figure 7. Machine D

It should be observed from the definition that if $\rho \geq \tau$ and if ρ is a two block partition then τ"r"ρ and τ"p"ρ are equivalent. If we realize a sequential machine coded into $\{0,1\}^n$ by h such that the g^{th} memory element is a set-reset flip-flop where $g \in \{1,\ldots,n\}$ then

$S_g(y,x) = \Phi_s(Y_g(y,x), y_g)$ and $R_g(y,x) = \varphi_r(Y_g(y,x), y_g)$ for all y in $\{0,1\}^n$ and for every input x. Φ_s and Φ_r are incomplete functions in that they are not specified for all evaluations of Y_g and y_g. Given a Boolean function $F:\{0,1\}^n \rightarrow \{0,1\}$ then we define $F^{max}(y) = F(y)$ if F(y) is specified and $F^{max}(y)=1$ if F(y) is not specified. We define $F^{min}(y) = F(y)$ if F(y) is specified and $F^{min}(y)=0$ otherwise. It should be noted that if F is extended to all of $\{0,1\}^n$ then $F^{max} \geq F \geq F^{min}$ where $F \geq F^{min}$ means that if $F^{min}(y)=1$ then $F(y)=1$. We are now ready for two lemmas.

Result 12 (Lemma).

Let M be a sequential machine coded by h into $\{0,1\}^n$. Let g be in $\{1,\ldots,n\}$ and let $\Lambda \leq \{1,\ldots,n\}$ with $g \notin \Lambda$. If $Y_g(y,x) = y_g U(y_\Lambda, x) + \bar{y}_g W(y_\Lambda, x)$ for every y in $\{0,1\}^n$ and for every input x where $W \leq U$ then M can be realized with the g^{th} memory element an r-s flip-flop such that $R_g(y,x) = I_g(y_\Lambda, x)$ while $S_g(y,x) = H_g(y_\Lambda, x)$ for every y in $\{0,1\}^n$ and for every input x.

Proof.

Since $S_g^{max}(y,x) = Y_g U(y_\Lambda, x) + \bar{y}_g W(y_\Lambda, x)$ and $S_g^{min}(y) = \bar{y}_g W(y_\Lambda, x)$ in general, $S_g(y,x) = \bar{y}_g V(y_\Lambda, x) + \bar{y}_g W(y_\Lambda, x)$ where V is any function such that $V \leq U$ for every y in $\{0,1\}^n$ and for every input x. Since $W \leq U$ we can let $V = W$. If this is done then $S_g(y,x) = W(y_\Lambda, x)$.

Similarly $R_g^{max}(y,x) = y_g\ \bar{U}(y_\Lambda,x) + \bar{y}_g\ \bar{W}(y_\Lambda,x)$ and $R_g^{min}(y,x) = y_g\ \bar{U}(y_\Lambda,x)$. Thus in general $R_g(y,x) = y_g\ \bar{U}(y_\Lambda,x) + \bar{y}_g\ \bar{M}(y_\Lambda,x)$ where $\bar{M}$ can be any function such that $\bar{M} \leq \bar{W}$. But $W \leq U$ implies that $\bar{W} \geq \bar{U}$. Thus we can let $\bar{M} = \bar{U}$. If this is done then $R_g(y,x) = U(y_\Lambda,x)$. ‖

The next Lemma is the converse of the previous one.

Result 13 (Lemma).

If M is coded by h into $\{0,1\}^n$ and realized such that the g^{th} memory element is a set-reset flip-flop with $S_g(y,x) = H_g(y_\Lambda,x)$ and $R_g(y,x) = I_g(y_\Lambda,x)$ for every y in $\{0,1\}^n$ and for every input x where Λ is a subset of $\{1,\ldots,n\}$ such that $g \notin \Lambda$. Then $Y_g(y,x) = y_g\ U(y_\Lambda,x) + \bar{y}_g\ W(y_\Lambda,x)$ where $U \geq W$.

Proof.

In general $Y_g(y,x) = y_g\ U(y,x) + \bar{y}_g\ W(y,x)$ and $S_g(y,x) = y_g\ V(y,x) + \bar{y}_g\ W(y,x)$ where $V \leq U$ while $R_g(y,x) = y_g\ \bar{U}(y,x) + \bar{y}_g\ \bar{M}(y,x)$ where $\bar{M} \leq \bar{W}$. Since $S_g(y,x) = H_g(y_\Lambda,x)$ then $V(y,x) = H_g(y_\Lambda,x) = W(y,x)$ since $g \notin \Lambda$. Since $R_g(y,x) = I_g(y_\Lambda,x)$ then $\bar{M}(y,x) = I_g(y_\Lambda,x) = \bar{U}(y,x)$. Thus $W(y,x) = H_g(y_\Lambda,x)$ and $U(y,x) = \bar{I}_g(y_\Lambda,x)$ which implies $W(y,x)$ can be written as $W(y_\Lambda,x)$ and $U(y,x)$ can be written as $U(y_\Lambda,x)$.

Show $W \leq U$. Since $V = W$, $\bar{M} = \bar{U}$, $V \leq U$ and $\bar{M} \leq \bar{W}$ we get immediately that $W \leq U$. ‖

With the aid of the previous lemmas we can now state and prove the theorems which give the necessary and sufficient conditions for a machine realized with set-reset flip-flops to have reduced dependence.

Result 14 (Theorem).

Let M be a machine coded by h into $\{0,1\}^n$ and realized such that the g^{th} memory element is a set-reset flip-flop where $g\in\{1,\ldots,n\}$. In addition for every y in $\{0,1\}^n$ and for every input x $S_g(y,x) = H_g(y_\Lambda,x)$ while $R_g(y,x) = I_g(y_\Lambda,x)$ where Λ is a subset of $\{1,\ldots,n\}$. Then the partitions $\rho = \rho_g$ and $\tau = \prod_\Lambda \rho_i$ are in relation τ"r"ρ.

Proof.

i) Suppose $g\in\Lambda$. As in Result 13 $Y_g(y,x) = y_g\, U(y,x) + \bar{y}_g\, W(y,x)$ while $S_g(y,x) = y_g\, V(y,x) + \bar{y}_g\, W(y,x)$ and $R_g(y,x) = y_g\, \bar{U}(y,x) + \bar{y}_g\, \bar{M}(y,x)$. Therefore multiplying S_g by $\bar{y}_g$ gives $\bar{y}_g\, S_g(y,x) = \bar{y}_g\, W(y,x)$ and multiplying $\bar{R}_g$ by y_g gives $y_g\, \bar{R}_g(y,x) = y_g\, U(y,x)$. Therefore $Y_g(y,x)$ $= y_g\, \bar{R}_g(y,x) + \bar{y}_g\, S_g(y,x)$. Which implies that $Y_g(y,x) = y_g\, \bar{I}_g(y_\Lambda,x)$ $+ \bar{y}_g\, H_g(y_\Lambda,x)$ from the hypothesis. Therefore $Y_g(y,x) = F_g(y_\Lambda,x)$ since $g\in\Lambda$ which from Result 2 implies $\tau\cdot\rho$"p"ρ. Since $g\in\Lambda$ this implies $\rho\geq\tau$ thus τ"p"ρ and since ρ is a 2 block partition τ"r"ρ.

ii) Suppose $g\notin\Lambda$. Then from Result 13 $Y_g(y,x) = y_g\, U(y_\Lambda,x)$ $+ \bar{y}_g\, W(y_\Lambda,x)$ with $U \geq W$. Thus from Result 2 again $\tau\cdot\rho$"p"ρ. Let a,b be in $\{s\}$ and x be an input such that $\delta(a,x)$ and $\delta(b,x)$ are specified.

Also let $\tau[a] = \tau[b]$ and $\rho[a] \neq \rho[b]$. Since $\tau[a] = \tau[b]$ and $\tau = \prod_{\Lambda} \rho_i$ we have that $h_\Lambda(a) = h_\Lambda(b)$. Since $\rho[a] \neq \rho[b]$ and $\rho = \rho_g$ we have that $h_g(a) \neq h_g(b)$.

Suppose $\delta(a,x) \in \rho[b]$. This means that $Y_g(h(a),x) = h_g(b)$ since $h_g(\delta(a,x)) = Y_g(h(a),x)$. Show $Y_g(h(a),x) = h_g(b)$. Suppose $h_g(b) = 0$ and $h_g(a) = 1$. Then since $Y_g(h(a),x) = h_g(a)\, U(h_\Lambda(a),x) + \bar{h}_g(a)\, W(h_\Lambda(a),x)$ we have that $Y_g(h(a),x) = U(h_\Lambda(a),x)$ which is 0. Since $U \geq W$ this implies $W(h_\Lambda(a),x) = W(h_\Lambda(b),x) = 0$. And since in this case $Y_g(h(b),x) = W(h_\Lambda(b),x) = 0$ we have $Y_g(h(b),x) = h_g(b)$. Suppose $h_g(b) = 1$ which implies $h_g(a) = 0$. Then $Y_g(h(a),x) = W(h_\Lambda(a),x) = 1$. Since $U \geq W$, this implies that $U(h_\Lambda(a),x) = U(h_\Lambda(b),x) = 1$. $Y_g(h(b),x) = U(h_\Lambda(b),x) = 1$ in this case. Thus $Y_g(h(b),x) = h_g(b)$. Since $Y_g(h(b),x) = h_g(b)$ we have that $\rho_g[\delta(b,x)] = \rho_g[b]$. Thus $\delta(a,x) \in \rho[b]$ implies $\delta(b,x) \in \rho[b]$. Similarly $\delta(b,x) \in \rho[a]$ implies $\delta(a,x) \in \rho[a]$. Therefore τ "r" ρ. ‖

The previous theorem has a corollary which tells when partition pairs relate to reduced dependence for set-reset realizations.

Result 15 (Corollary).

If a sequential machine coded by h into $\{0,1\}^n$ has a realization with set-reset flip-flops such that $R_g(y,x) = I_g(y_\Lambda,x)$ and $S_g(y,x) = H_g(y_\Lambda,x)$ for all g in Λ' where Λ' and Λ are subsets of $\{1,\ldots,n\}$

such that $\Lambda' \leq \Lambda$, then the partitions $\tau = \prod_{\Lambda} \rho_i$ and $\rho = \prod_{\Lambda'} \rho_i$ are such that τ"p"ρ.

Proof.

From Result 14 τ"r"ρ_g for every g in Λ'. Since $g \epsilon \Lambda$ we have that $\rho_g \geq \tau$ and therefore τ"p"ρ_g. From Result 1 this implies τ"p"$\prod_{\Lambda'} \rho_g$ since the multiplication of partition pairs is a partition pair. Therefore τ"p"ρ. ‖

It should be noted that if $\Lambda' = \Lambda$ in Result 15 that τ has S.P. Next we state the sufficiency theorem for reduced dependence.

Result 16 (Theorem).

Let a sequential machine coded by h into $\{0,1\}^n$ have state partitions τ and ρ such that τ"r"ρ. If in addition $\tau = \prod_{\Lambda} \rho_i$ and $\rho = \rho_g$ where $g \epsilon \{1, \ldots, n\}$ and $\Lambda \leq \{1, \ldots, n\}$ then $R_g(y,x) = I_g(y_\Lambda, x)$ and $S_g(y,x) = H_g(y_\Lambda, x)$ for every y in $\{0,1\}^n$ and for every input x.

Proof.

i) Suppose $g \epsilon \Lambda$. Then $\rho \geq \tau$ and since τ"r"ρ implies $\tau \cdot \rho$"p"ρ we get τ"p"ρ since $\tau \cdot \rho = \tau$. Therefore $Y_g(y,x) = F_g(y_\Lambda, x)$ or $Y_g(y,x) = y_g M(y_\Gamma, x) + \bar{y}_g N(y_\Gamma, x)$ where $\Gamma = \Lambda - \{g\}$. As in the proof of Result 12 $R_g(y,x) = y_g \bar{U}(y_\Gamma, x) + \bar{y}_g \bar{M}(y_\Gamma, x)$ and $S_g(y,x) = y_g V(y_\Gamma, x) + \bar{y}_g W(y_\Gamma, x)$.

ii) Suppose $g \notin \Lambda$. τ "r" ρ implies $\tau \cdot \rho$ "p" ρ. Therefore $Y_g(y,x) = F_g(y_\Lambda, y_g, x)$ from Result 2. Again we have certain freedoms on F_g. Namely for every $a \in \{s\}$ such that $\delta(a,x)$ is specified $F_g(h_\Lambda(a), h_g(a), x) = h_g[\delta(a,x)]$. Otherwise we may specify F_g in any manner. We specify F_g on the (y_Λ, y_g) such that $(y_\Lambda, y_g) \neq (h_\Lambda(a), h_g(a))$ for any $a \in \{s\}$ with $\delta(a,x)$ specified as follows. If $y_\Lambda(a) = h_\Lambda(a)$ for some $a \in \{s\}$ but $y_g \neq h_g(a)$, then define $F_g(y_\Lambda, y_g, x) = F_g(h_\Lambda(a), h_g(a), x)$. For the (y_Λ, y_g) that remain we define F_g with the constraint that $F_g(y_\Lambda, y_g, x) = F_g(y_\Lambda, \bar{y}_g, x)$ which can clearly be done.

Since $Y_g(y,x) = F_g(y_\Lambda, y_g, x)$, it is clear that $Y_g(y,x) = y_g U(y_\Lambda, x) + \bar{y}_g W(y_\Lambda, x)$. Show $U \geq W$. Let $v_\Lambda \in \{0,1\}^\Lambda$. Clearly there exists $y, y' \in \{0,1\}^n$ such that $v_\Lambda = y_\Lambda = y'_\Lambda$ but $y_g \neq y'_g$. Suppose with no loss of generality that $y_g = 1$, $y'_g = 0$. Then $Y_g(y,x) = F_g(y_\Lambda, y_g, x) = U(y_\Lambda, x)$ and $Y_g(y',x) = F_g(y'_\Lambda, y_g, x) = W(y'_\Lambda, x)$. If either of y or y' is in $\{0,1\}^n - h(\{s\})$ then $F_g(y'_\Lambda, y'_g, x) = F(y_\Lambda, y_g x)$ from the way we specified F_g. Note that $y \in \{0,1\}^n - h(\{s\})$ means $y \neq h(a)$ for every $a \in \{s\}$. $F_g(y'_\Lambda, y'_g, x) = F_g(y_\Lambda, y_g, x)$ implies that $U(y_\Lambda, x) = W(y'_\Lambda, x)$ and therefore $U(v_\Lambda, x) = W(v_\Lambda, x)$. Suppose both of y, y' are in $h(\{s\})$. Then $y = h(a)$, $y' = h(b)$. Again suppose $y'_g = 0$ and $y_g = 1$. If $W(v_\Lambda, x) = 1$ then $W(y'_\Lambda, x) = 1$ which implies that $Y_g(y', x) = F_g(y'_\Lambda, y_g, x) = 1$. This means that $Y_g(h(b), x) = h_g(\delta(b,x)) = 1$. Hence $\delta(b,x) \in \rho_g[a]$

since $h_g(a) = y_g = 1$. From the hypothesis $h_\Lambda(a) = h_\Lambda(b)$ implies that $\tau[a] = \rho[b]$ and $h_g(a) \neq h_g(b)$ implies that $\rho[a] \neq \rho[b]$. Since τ"r"ρ and $\delta(b,x) \in \rho_g[a]$, it must be true that $\delta(a,x) \in \rho_g[a]$. This means that $Y_g(h(a),x) = h_g[\delta(a,x)] = h_g(a) = 1$. Thus $1 = Y_g(y,x) = U(y_\Lambda, x)$ $= U(v_\Lambda, x)$. Thus we have shown that $U \geq W$. From Result 13 this implies the theorem. ‖

Machine D of Figure 7 can be used to illustrate Result 17. It has been shown that $(\overline{1,2,3,4;5})$"r"$(\overline{1,2;3,4,5})$. If machine D is coded such that $\rho_1 = (\overline{1,2,3,4;5})$ and $\rho_2 = (\overline{1,2;3,4,5})$ then from Result 17 $R_2(y_1, y_2, x) = I_2(y_1, x)$ and $S_2(y_1, y_2, x) = H_2(y_1, x)$ for every (y_1, y_2) $\in \{0,1\}^2$ and for every input x. For one particular coding $R_2(y_1, y_2, x)$ $= x\bar{y}_1$, and $S_2(y_1, y_2, x) = \bar{y}_1 \bar{x}$.

We now consider some of the properties of the relation "r".

Result 17.

If τ"r"ρ and $\gamma \leq \tau$ then γ"r"ρ.

Proof.

i) Since $\gamma \leq \tau$, we infer that $\gamma \cdot \rho \leq \tau \cdot \rho$. From τ"r"ρ we know that $\tau \cdot \rho$"p"ρ and from Result 1 this implies that $\gamma \cdot \rho$"p"ρ.

ii) Let $a, b \in \{s\}$ and x be an input such that $\gamma[a] = \gamma[b]$ and $\rho[a] \neq \rho[b]$ with $\delta(a,x)$ and $\delta(b,x)$ specified. Since $\tau \geq \gamma$ this implies $\tau[a] = \tau[b]$. From τ"r"ρ this implies that if $\delta(a,x) \in \rho[b]$ then $\delta(b,x) \in \rho[b]$. ‖

Result 18.

If τ_1"r"ρ and τ_2"r"ρ then $\tau_1 \cdot \tau_2$"r"ρ.

Proof.

This result follows from Result 17 since $\tau_1 \cdot \tau_2 \leq \tau_1$.

Result 19.

If τ and ρ are partitions such that τ"p"ρ and $\rho \geq \tau$ then τ"t"ρ_1 and τ"r"ρ_1 where ρ_1 is any 2 block partition such that $\rho_1 \geq \rho$.

Proof.

Since τ"p"ρ and $\rho_1 \geq \rho$, we know that τ"p"ρ_1. Since $\rho_1 \geq \tau$ we deduce that τ"t"ρ_1 and τ"r"ρ_1.

In general τ_1"r"ρ and τ_2"r"ρ does not imply $(\tau_1 + \tau_2)$"r"ρ. This means that for a given partition ρ there does not necessarily exist a largest partition τ such that τ"r"ρ. This is shown by Machine E of Figure 8. It is true that if either $\tau_1 \leq \rho$ or $\tau_2 \leq \rho$ where τ_1"r"ρ and τ_2"r"ρ then $(\tau_1 + \tau_2)$"r"ρ when the machine is completely specified. The next result in this section is concerned with the conditions such that more than one flip-flop can have reduced dependence.

		Inputs 0	1
	1	3	1
	2	4	2
States	3	5	4
	4	2	4
	5	5	3

$\overline{(1,2,4,5;3)}$"r"$\overline{(1,2,3,4;5)}$

$\overline{(1,2,4;3,5)}$"r"$\overline{(1,2,3,4,5)}$

$\overline{(1,2,3,4,5)}$"not r"$\overline{(1,2,3,4;5)}$

Figure 8. Machine E

Result 20 (Theorem).

Let M be a machine coded by h into $\{0,1\}^n$ such that

i) There exists state partitions in the r-relation as follows: τ_1"r"ρ_1, τ_2"r"$\rho_2, \ldots, \tau_n$"r"ρ_n where ρ_i is the partition associated with h_i.

ii) For every i in $\{1,\ldots,n\}$ there exists $\Lambda_i \leq \{1,\ldots,n\}$ such that $\tau_i \geq \prod_{\Lambda_i} \rho_j$.

Then there is a realization with set-reset flip-flops such that for every i in $\{1,\ldots,n\}$ $R_i(y,x) = I_i(y_{\Lambda_i}, x)$ and $S_i(y,x) = H_i(y_{\Lambda_i}, x)$ for every y in $\{0,1\}^n$ and for every input x.

Proof.

From Result 17 and i) of the hypothesis $\prod_{\Lambda_i} \rho_j$"r"ρ_i for every i in $\{1,\ldots,n\}$. The theorem now follows from Result 16. ‖

The converse to Result 21 is also obviously true from Result 15. In the remainder of the section we want to prove that partition pairs are sufficient but not necessary for reduced dependence when a machine is realized using set-reset flip-flops. We do this by defining a relation "$\bar{r}$" which is the same as the relation "r" except when τ"$\bar{r}$"ρ it is not required that ρ be a 2 block partition.

Definition 9. State partitions τ and ρ are in relation τ"$\bar{r}$"ρ iff τ and ρ satisfy 2 and 3 of Definition 8.

Note that if τ"p"ρ then τ"$\bar{r}$"ρ. Figure 9 gives an example of the relation $\bar{r}$. Let $\tau = \overline{(1,3;\overline{2,4,5})}$ and $\rho = \overline{(1,2;\overline{3,4;5})}$. Since $\tau \cdot \rho = \overline{(1;2;3;4;5)} = \emptyset$ clearly $\tau \cdot \rho$"p"ρ. Also $\delta(1,0) \in \rho[1]$, $\delta(3,0) \in \rho[3]$, $\delta(2,0) \in \rho[2]$, $\delta(4,0) \in \rho[4]$, $\delta(5,0) \in \rho[5]$, $\delta(1,1) \in \rho[1]$, $\delta(3,1) \in \rho[3]$ while $\rho[\delta(2,1)] = \rho[\delta(4,1)] = \rho[\delta(5,1)]$. Hence τ"$\bar{r}$"ρ.

		Inputs 0	1
	1	2	1
	2	1	3
States	3	4	4
	4	3	4
	5	5	3

$\overline{(1,3;\overline{2,4,5})}$"$\bar{r}$"$\overline{(1,2;\overline{3,4;5})}$

Figure 9. Machine F

It is convenient to have the following properties.

Result 21.

If τ and ρ are such that τ"$\bar{r}$"ρ and if $\tau_1 \leq \tau$, then τ_1"$\bar{r}$"ρ.

Proof.

Identical to the proof of Result 17.

Result 22.

If τ"$\bar{r}$"ρ and $\rho_1 \geq \rho$, then τ"$\bar{r}$"ρ_1.

Proof.

i) Let $a,b \in \{s\}$ such that $\tau[a] = \tau[b]$ and $\rho_1[a] = \rho_1[b]$ and let $x \in \{x\}$ such that $\delta(a,x)$ and $\delta(b,x)$ are specified. If $\rho[a] = \rho[b]$ then $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ since $\tau \cdot \rho$"p"ρ. This implies that $\rho_1[\delta(a,x)] = \rho_1[\delta(b,x)]$ since $\rho_1 \geq \rho$. If $\rho[a] \neq \rho[b]$ then either $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ which again implies $\rho_1[\delta(a,x)] = \rho_1[\delta(b,x)]$ or $\delta(a,x) \in \rho[a]$ and $\delta(b,x) \in \rho[b]$. Since $\rho_1[a] = \rho_1[b]$ this implies $\delta(a,x) \in \rho_1[a]$ and $\delta(b,x) \in \rho_1[b]$. Thus $\tau \cdot \rho$"p"ρ.

ii) Let $a,b \in \{s\}$ such that $\tau[a] = \tau[b]$ and $\rho_1[a] \neq \rho_1[b]$ and let $x \in \{x\}$ such that $\delta(a,x)$ and $\delta(b,x)$ are specified. Since $\rho_1 \geq \rho$ we have that $\rho[a] \neq \rho[b]$ and therefore either $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ which implies that $\rho_1[\delta(a,x)] = \rho_1[\delta(b,x)]$; that is, $\delta(a,x) \in \rho[a]$ and $\delta(b,x) \in \rho[b]$. Since $\rho_1 \geq \rho$ this implies that $\delta(a,x) \in \rho_1[a]$ and $\delta(b,x) \in \rho_1[b]$. ‖

The next result is the principal result concerning $\bar{r}$.

Result 23 (Theorem).

If a machine is coded by h into $\{0,1\}^n$ such that τ "$\bar{r}$" ρ where $\tau = \prod_{\Lambda} \rho_i$ and $\rho = \prod_{\Lambda'} \rho_i$ with $\Lambda' < \{1,\ldots,n\}$ and $\Lambda \leq \{1,\ldots,n\}$ then $R_g(y,x) = I_g(y_\Lambda, x)$ and $S_g(y,x) = H_g(y_\Lambda, x)$ for every $g \epsilon \Lambda'$ where $y \epsilon \{0,1\}^n$ and x is an input.

Proof.

Let $g \epsilon \Lambda'$. Then since $\rho_g \geq \rho$ and τ "$\bar{r}$" ρ we deduce that τ "$\bar{r}$" ρ_g from Result 22. Since ρ_g is a 2 block partition this means τ "r" ρ_g. From Result 16 this implies the result.

Result 24 (Corollary).

If a machine is coded by h into $\{0,1\}^n$ such that τ "p" ρ where $\tau = \prod_{\Lambda} \rho_i$ and $\rho = \prod_{\Lambda'} \rho_i$ with $\Lambda' \leq \{1,\ldots,n\}$ and $\Lambda \leq \{1,\ldots,n\}$ then $R_g(y,x) = I_g(y_\Lambda, x)$ and $S_g(y,x) = H_g(y_\Lambda, x)$ for every $g \epsilon \Lambda'$.

Proof.

τ "p" ρ implies τ "$\bar{r}$" ρ which from Result 23 implies the result.

Result 24 says the partition pairs are sufficient for reduced dependence when a machine is realized using set-reset flip-flops. The following example shows they are not necessary by showing that the relation "$\bar{r}$" is not necessary.

Consider Machine G in Figure 10. If Machine G is coded by h into $\{0,1\}^3$ such that $\rho_1 = (\overline{1,3,5;2,4})$, $\rho_2 = (\overline{1,2,3,4;5})$ and $\rho_3 = (\overline{1,2;3,4,5})$ then we know from Result 16 that S_1, R_1, S_2 and R_2 are functions of only y_2 and the input for every $(y_1, y_2, y_3) \in \{0,1\}^3$. But the partition $(\overline{1,2;3,4,5})$ is not in relation $\bar{r}$ with $(\overline{1,3;2,4;5})$ which equals $\rho_1 \cdot \rho_2$. Thus "$\bar{r}$" is not necessary for reduced dependence.

		Inputs 0	1	
	1	2	1	$(\overline{1,2;3,4,5})$"r"$(\overline{1,3,5;2,4})$
	2	4	2	$(\overline{1,2;3,4,5})$"r"$(\overline{1,2,3,4;5})$
States	3	1	3	$(\overline{1,2;3,4,5})$"not $\bar{r}$"$(\overline{1,3;2,4;5})$
	4	3	4	
	5	5	1	

Figure 10. Machine G

In this chapter we have shown that partition pairs are neither necessary nor sufficient for reduced dependence of the input functions to trigger flip-flops. It has also been shown that partition pairs are sufficient but not necessary for reduced dependence of the input function to set-reset flip-flops. In order to analize a machine completely with respect to trigger or set-reset type realizations it is necessary to consider all two block partitions of $\{s\}$. There are $2^{q-1}-1$ of them for a q state machine. For each of these partitions ρ one has a set of

partitions $\bar{\tau}(\rho)$ where the elements of $\bar{\tau}(\rho)$ are partitions τ such that τ"t"ρ. For the r relation there is a set of partitions $\bar{\gamma}(\rho)$, whose elements are partitions γ such that γ"r"ρ. In the general situation this is as much as one can say. If the machine is completely specified, then for each ρ there is a largest partition τ such that τ"t"ρ. Thus in this case we need only consider each ρ and the largest partition τ such that τ"r"ρ. If one considers partition pairs, then one can compute the Mm pairs (Reference 1) and the Mm pairs imply all other partition pairs. Thus it is clear that one cannot store the information regarding the t-pairs and r-pairs as compactly as one can for the partition pairs. It should however be noted that the computation of the Mm pairs of Reference 1 is a difficult task in general.

CHAPTER 3

Feedback

Introduction

In previous studies of feedback in sequential machines (Reference 3) the machine was considered to be realized using unit delay memory elements. In this chapter we study feedback in sequential machines which are realized with trigger or set-reset flip-flop memory elements. It is shown that the different memory elements affect the feedback characteristics of a sequential machine. We begin by restating some results given in Reference 3.

Definition 10. Let $M = (\{s\}, \{x\}, \{0\}, \delta, \lambda)$ be a sequential machine. Let τ and ρ be state partitions and let $f: \{s\} \times \{x\} \rightarrow D$ be some function where D is a set. Then τ "pf" ρ iff for every two states a,b such that $\tau[a] = \tau[b]$ and for every input x such that $\delta(a,x)$ and $\delta(b,x)$ are specified and $f(a,x) = f(b,x)$ then $\rho[\delta(a,x)] = \rho[\delta(b,x)]$.

The next theorem relates the relation pf to unit delay realizations.

Result 25 (Theorem).

Let M be a machine coded by h into $\{0,1\}^n$. Also let $\tau = \prod_{\Lambda} \rho_i$ where $\Lambda \leq \{1,\ldots,n\}$ and let $\rho = p_g$ where $g \in \{1,\ldots,n\}$. Then τ "pf" ρ iff $Y_g(h(a),x) = F_g(h_\Lambda(a), f(a,x), x)$ when $\delta(a,x)$ is specified.

Proof.

i) Suppose τ "pf" ρ. Define F_g as follows. For every (y_Λ, d, x) where $d \in D$ and $y_i \in \{0,1\}$ for every i in Λ let $F_g(y_\Lambda, d, x) = h_g[\delta(a,x)]$ if there exists a $\in \{s\}$ such that $\delta(a,x)$ is specified and $(h_\Lambda(a), f(a,x)) = (y_\Lambda, d)$. Show F_g is well defined. If there exists $a,b \in \{s\}$ such that $\delta(a,x)$ and $\delta(b,x)$ are specified, $\tau[a] = \tau[b]$ and $f(a,x) = f(b,x)$ then $\rho_g[\delta(a,x)] = \rho_g[\delta(b,x)]$ which implies $h_g[\delta(a,x)] = h_g[\delta(b,x)]$ hence F_g is well defined. For all (y_Λ, d, x) such that there is no $a \in \{s\}$ and input x such that $(h_\Lambda(a), f(a,x)) = (y_\Lambda, d)$ then $F_g(y_\Lambda, d, x)$ can be specified in any manner.

ii) Suppose $h_g[\delta(a,x)] = F_g(h_\Lambda(a), f(a,x), x)$. Let $a,b \in \{s\}$ such that $\delta(a,x)$ and $\delta(b,x)$ are specified, $\tau[a] = \tau[b]$ and $f(a,x) = f(b,x)$. Since $\tau[a] = \tau[b]$ implies $h_i(a) = h_i(b)$ for every $i \in \Lambda$ this implies from the hypothesis that $h_g[\delta(a,x)] = h_g[\delta(b,x)]$ which implies $\rho[\delta(a,x)] = \rho[\delta(b,x)]$. ||

Definition 11. Let M be a machine and τ a state partition of M then $m^1_{pf}(\tau) = \prod(\rho_i | \tau$ "pf" $\rho)$ and $m_{pf}^{i+1}(\tau) = m^1_{pf}(m^i_{pf}(\tau))$.

Our m_{pf} is the same as the m operator of Hartmanis and Stearns. We now give the result of Hartmanis and Stearns on feedback which we shall not prove. We do not use this result except to relate flip-flop realizations to delay realizations.

Result 26 (Theorem).

Let M be a q state sequential machine and let $f:\{s\} \times \{x\} \rightarrow D$. Then M can be realized using f for feedback iff $m_{pf}^{q-1}(I) = \emptyset$. I is the unit partitions.

Feedback and Trigger Flip-Flop Realizations

In order to determine when a function f can be used as feedback in a machine M realized with trigger flip-flops we define the following relation. The definition of what we mean by the expression "using f for feedback" will be given later.

Definition 12. Let τ and ρ be state partitions on machine M and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$ where ℓ is a positive integer. Then τ "tf" ρ iff

1. ρ is a two block partition
2. $\tau \cdot \rho$ "pf" ρ
3. For every 2 states a,b and for every input x such that $\delta(a,x)$ and $\delta(b,x)$ are specified, $\tau[a] = \tau[b]$, $\rho[a] \neq \rho[b]$, and $f(a,x) = f(b,x)$ then $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$.

If we have a machine M and a function $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$ and if we code M with h into $\{0,1\}^{n}$, then we can define a function f' on $\{0,1\}^{n}$ by $f'(h(a),x) = f(a,x)$. If h is not onto $\{0,1\}^{n}$ then we extend f' in any manner to all of $\{0,1\}^{n}$. We make no distinction between

f and f'. It might be noted that we use the set $\{0,1\}^{\ell}$ rather than D for the range of f. We do this because we are interested in a realization which can be realized in a practical since. In other words the output of f will be fed into logical gates, hence it is convenient to consider the outputs as n tuples of $\{0,1\}$. It is necessary to prove the next two results before we can characterize feedback in trigger flip-flop realizations.

Result 27 (Lemma).

Let machine M be realized with the g^{th} memory element a trigger flip-flop where $h:\{s\} \rightarrow \{0,1\}^{n}$ is the coding function and $g \in \{1,\ldots,n\}$. If

i) $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$ and $\Lambda \leq \{1,\ldots,n\}$ with $g \notin \Lambda$.

ii) $F_g(y,x) = y_g M(y_\Lambda, f(y,x),x) + \bar{y}_g N(y_\Lambda, f(y,x),x)$ for every $x \in \{x\}$ and $y \in \{0,1\}^{n}$ where $M = \bar{N}$.

Then $T_g(y,x) = G_g(y_\Lambda, f(y,x),x)$.

Proof.

Since $T_g(y,x) = y_g \bar{Y}_g(y,x) + \bar{y}_g Y_g(y,x)$ for every $y \in \{0,1\}^{n}$ and for every $x \in \{x\}$ if we substitute for Y_g and simplify we get that $T_g(y,x) = y_g \bar{M}(y_\Lambda, f(y,x),x) + \bar{y}_g N(y_\Lambda, f(y,x),x)$. Since $M = \bar{N}$ this implies $T_g(y,x) = N(y_\Lambda, f(y,x),x)$. ‖

Result 28 (Lemma).

Let machine M be coded by h into $\{0,1\}^n$. Let $\tau = \prod_{\Lambda} \rho_i$ where $\Lambda \leq \{1,\ldots,n\}$, let $\rho = \rho_g$ where $g \in \{1,\ldots,n\}$ and let $f:\{s\} \times \{x\} \to \{0,1\}^{\ell}$. If τ "tf" ρ then $T_g(y,x) = G_g(y_\Lambda, f(y,x),x)$.

Proof.

Since τ "tf" ρ we know that $\tau \cdot \rho$ "pf" ρ and from Result 25 this implies that $Y_g(y,x) = F_g(y_\Lambda, y_g, f(y,x),x)$ for every y in $\{0,1\}^n$.

i) Suppose $g \in \Lambda$. Then since $T_g(y,x) = y_g \bar{Y}_g(y,x) + \bar{y}_g Y_g(y,x)$ we deduce that $T_g(y,x) = y_g \bar{F}_g(y_\Lambda, y_g, f(y,x),x) + \bar{y}_g F_g(y_\Lambda, y_g, f(y,x),x)$ $= G_g(y_\Lambda, f(y,x),x)$.

ii) Suppose $g \notin \Lambda$. Then $Y_g(y,x) = y_g M(y_\Lambda, f(y,x),x)$ $+ \bar{y}_g N(y_\Lambda, f(y,x),x)$ where $M(y_\Lambda,d,x) = F_g(y_\Lambda, y_g = 1,d,x)$ for every $d \in \{0,1\}^{\ell}$ and $N(y_\Lambda,d,x) = F_g(y_\Lambda, y_g = 0,d,x)$. Recall that there are certain freedoms on F_g. Namely $F_g(h_\Lambda(a), h_g(a), f(h(a),x),x) = h_g[\delta(a,x)]$ if $\delta(a,x)$ is specified. Otherwise $F_g(y_\Lambda, y_g, d,x)$ where $d \in \{0,1\}^{\ell}$ is arbitrary. Specify F_g as follows. If $F_g(y_\Lambda, y_g, d,x)$ is not determined as above, let $F_g(y_\Lambda, y_g, d,x) = \bar{F}_g(y_\Lambda, \bar{y}_g, d,x)$ where $d \in \{0,1\}^{\ell}$, $y \in \{0,1\}^n$ and $x \in \{x\}$. Show $F_g(y_\Lambda, y_g, d,x) \neq F_g(y_\Lambda, \bar{y}_g, d,x)$ for all $y \in \{0,1\}^n$, $x \in \{x\}$ and $d \in \{0,1\}^{\ell}$. We must only consider the case where there exists $a,b \in \{s\}$, $x \in \{x\}$ such that $h_\Lambda(a) = h_\Lambda(b)$, $h_g(a) \neq h_g(b)$, $f(a,x)$ $= f(b,x)$ and $\delta(a,x)$ and $\delta(b,x)$ are specified. In this case

$F_g(h_\Lambda(a), h_g(a), f(h(a),x),x) = h_g[\delta(a,x)]$. But $h_\Lambda(a) = h_\Lambda(b)$ implies that $\tau[a] = \tau[b]$ and $h_g(a) \neq h_g(b)$ implies that $\rho[a] \neq \rho[b]$. Since $f(a,x) = f(b,x)$, $\delta(a,x)$ and $\delta(b,x)$ are specified and τ"tf"ρ, this implies that $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ and therefore that $h_g[\delta(a,x)] \neq h_g[\delta(b,x)]$. Thus $F_g(h_\Lambda(a), h_g(a), f(h(a),x),x) \neq F_g(h_\Lambda(b), h_g(b), f(h(b),x),x)$.

Show $M = \bar{N}$ if F_g is so specified. This is clear since $M(y_\Lambda, d, x) = F_g(y_\Lambda, y_g = 1, d, x) \neq F_g(y_\Lambda, y_g = 0, d, x) = N(y_\Lambda, d, x)$ for every y_Λ with $y_i \in \{0,1\}^\ell$ for every $i \in \Lambda$, $x \in \{x\}$ and $d \in \{0,1\}^\ell$. From Result 27 this implies the theorem. ‖

Result 29 (Theorem).

If machine M is coded by h into $\{0,1\}^n$ and realized with trigger flip-flop memory elements such that $T_g(y,x) = G_g(y_\Lambda, f(y,x),x)$ where $\Lambda \leq \{1,\ldots,n\}$ and $g \in \{1,\ldots,n\}$ and $f:\{s\} \times \{x\} \to \{0,1\}^\ell$, then τ"tf"ρ where $\tau = \prod_\Lambda \rho_i$ and $\rho = \rho_g$.

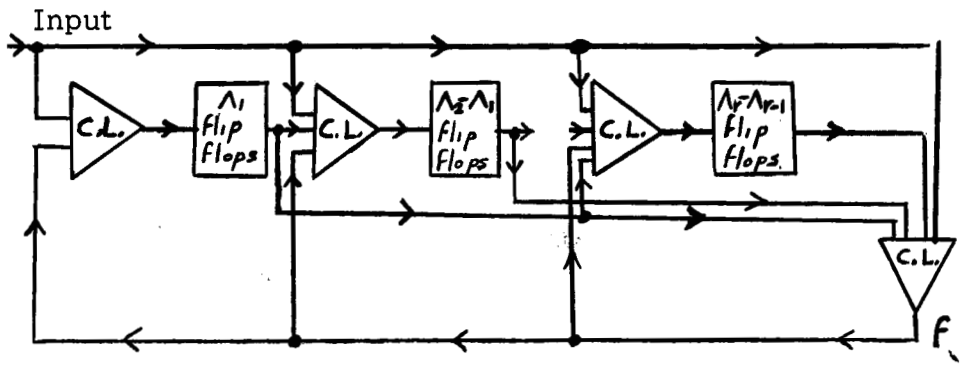
Proof.

i) Suppose $g \in \Lambda$. Then τ"tf"ρ is equivalent to τ"pf"ρ since $\rho \geq \tau$. In general $Y_g(y,x) = y_g \bar{T}_g(y,x) + \bar{y}_g T(y,x)$ thus $Y_g(y,x) = y_g \bar{G}_g(y_\Lambda, f(y,x),x) + \bar{y}_g G_g(y_\Lambda, f(y,x),x) = F_g(y_\Lambda, f(y,x),x)$. Therefore from Result 25 τ"pf"ρ.

ii) Suppose $g \notin \Lambda$. Then again $Y_g(y,x) = y_g \bar{G}_g(y_\Lambda, f(y,x),x) + \bar{y}_g G_g(y_\Lambda, f(y,x),x) = F_g(y_\Lambda, y_g, f(y,x),x)$ and therefore from Result 25

$\tau \cdot \rho$"pf"ρ. Let $a,b \in \{s\}$ and $x \in \{x\}$ such that $\tau[a] = \tau[b]$, $\rho[a] \neq \rho[b]$, $f(a,x) = f(b,x)$, and $\delta(a,x)$ and $\delta(b,x)$ are specified. $\tau[a] = \tau[b]$ implies that $h_i(a) = h_i(b)$ for every $i \in \Lambda$ and $\rho[a] \neq \rho[b]$ implies that $h_g(a) \neq h_g(b)$. Assume $h_g(a) = 1$ which implies $h_g(b) = 0$. Then $Y_g(h(a),x) = \bar{G}_g(h_\Lambda(a), f(h(a),x),x)$ and $Y_g(h(b),x) = G_g(y_\Lambda, f(y,x),x)$. But since $(h_\Lambda(a), f(h(a),x),x) = (h_\Lambda(b), f(h(b),x),x)$, this implies that $Y_g(h(a),x) = \bar{Y}_g(h(b),x)$. Since $\delta(a,x)$ and $\delta(b,x)$ are specified, this means that $h_g[\delta(a,x)] \neq h_g[\delta(b,x)]$ which implies that $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$. The same argument yields the same result assuming $h_g(a) = 0$ which implies $h_g(b) = 1$. Therefore τ"tf"ρ. ||

With these results out of the way we can consider feedback in machines realized with trigger-flip-flop memory elements. First we must define this concept. The basic idea is to lay the machine out from left to right in such a way that the input function to the ith flip-flop can be computed from f and the state of that portion of the machine which lies to the left of the ith flip-flop. This is shown in Figure 11. It should be noted in the figure that the set of $\Lambda_r - \Lambda_{r-1}$ flip-flops consists of those flip-flops i such that $i \in \Lambda_r - \Lambda_{r-1}$.



C.L. denotes combinational logic
$\Lambda_1 < \Lambda_2 < \cdots < \Lambda_r \leq \{1, \ldots, n\}$

Figure 11.

Definition 13. Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Then M can be realized with trigger flip-flop memory elements using f for feedback iff M can be coded by h into $\{0,1\}^n$ such that

1. There exists $\{\Lambda_1, \ldots, \Lambda_k\}$ a set of positive integers such that $u < v$ implies $\Lambda_u < \Lambda_v$.

2. If $i \in \Lambda_1$ then $T_i(y,x) = G_i(f(y,x),x)$ for every $y \in \{0,1\}^n$ and $x \in \{x\}$.

3. If $i \in \Lambda_r - \Lambda_{r-1}$ where $1 < r \leq k$ $T_i(y,x) = G_i(y_{\Lambda_{r-1}}, f(y,x), x)$

4. $\prod_{\Lambda_k} \rho_i = \emptyset$ where ρ_i is the partition associated with h_i.

It should be noticed that our definition of feedback depends on the memory element used. In order to prove the important theorem of

this section we define the following quantity and prove a property about it.

Definition 14. Let M be a machine and τ be a state partition of M. Then $m^1_{tf}(\tau) = \Pi\{\rho \mid \tau \text{ "tf" } \rho\}$ and $m^{i+1}_{tf}(\tau) = m^1_{tf}(m^i_{tf}(\tau))$ for every integer i. If $\{\rho \mid \tau \text{ "tf" } \rho\} = \varphi$ the empty set we define $m^1_{tf}(\tau) = I$ the unit partition.

We frequently designate $m^1_{tf}(\tau)$ by $m_{tf}(\tau)$. It should be observed that τ and $m_{tf}(\tau)$ are not in the relation "tf".

Result 30 (Lemma).

If $\tau \cdot \tau_1$ and ρ are state partitions such that $\tau_1 \leq \tau$ and τ "tf" ρ where $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$ then τ_1 "tf" ρ.

Proof.

Let $\tau_1[a] = \tau_1[b]$. Then $\tau[a] = \tau[b]$ since $\tau \geq \tau_1$. If $\rho[a] = \rho[b]$, $f(a,x) = f(b,x)$ and $\delta(a,x)$ and $\delta(b,x)$ are specified then $\rho[\delta(a,x) = \rho[\delta(b,x)]$ since $\tau \cdot \rho$ "pf" ρ. If $\rho[a] \neq \rho[b]$, $f(a,x) = f(b,x)$ and $\delta(a,x)$ and $\delta(b,x)$ are specified then $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ since τ "tf" ρ. ‖

Result 31.

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. If τ and τ_1 are state partitions such that $\tau_1 \leq \tau$ then $m_{tf}(\tau_1) \leq m_{tf}(\tau)$.

Proof.

Let ρ be such that τ "tf" ρ then τ_1 "tf" ρ from Result 30. This implies $m_{tf}(\tau_1) \leq m_{tf}(\tau)$. ‖

Result 32.

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Then $m_{tf}^{i+1}(I) \leq m_{tf}^{i}(\tau)$ for every $i \in \{1,2,\ldots\}$.

Proof.

i) Show $m_{tf}^{2}(I) \leq m_{tf}^{1}(I)$. Clearly $m_{tf}^{1}(I) \leq I$. Thus $m_{tf}[m_{tf}^{1}(I)] \leq m_{tf}(I)$ from Result 31 which implies $m_{tf}^{2}(I) \leq m_{tf}(I)$.

ii) Suppose $m_{tf}^{i+1}(I) \leq m_{tf}^{i}(I)$. Then $m_{tf}[m_{tf}^{i+1}(I)] \leq m_{tf}[m_{tf}^{i}(I)]$ from Result 31 and therefore $m_{tf}^{i+2}(I) \leq m_{tf}^{i+1}(I)$.

It is clear that if $m_{tf}^{i}(I) = m_{tf}^{i+1}(I)$ then $m_{tf}^{i+2}(I) = m_{tf}^{i}(I)$. Therefore since I can be refined at most q-1 times if M is a q state machine we know that $m_{tf}^{q-1}(I) = m_{tf}^{q}(I)$.

It should be noticed that $m_{tf}(\tau)$ is a fairly difficult quantity to calculate. At this point we must consider every two block partition ρ and see if τ "tf" ρ and multiply these ρ together. Later we will give a better method. But first we prove a major result and then give an example of some of these concepts.

Result 33 (Theorem).

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. M can be realized

with trigger flip-flops using f for feedback iff $m_{tf}^{q-1}(I) = \emptyset$ where q is the number of states of M.

Proof.

Suppose M can be realized with trigger flip-flops using f for feedback. Then M can be coded by h into $\{0,1\}^n$ such that Definition 13 is satisfied. Let $\tau_r = \prod_{\Lambda_r} \rho_j$.

1. From 2 of Definition 13 and Result 29 $I"tf"\rho_i$ when $i \in \Lambda_r$ and ρ_i is the partition associated with h_i. This implies that $m_{tf}^1(I) \leq \prod_{\Lambda_i} \rho_j$ and $m_{tf}(I) \leq \tau_1$.

2. From 3 of Definition 13 and Result 29 $\prod_{\Lambda_{r-1}} \rho_j"tf"\rho_i$ for every $i \in \Lambda_r - \Lambda_{r-1}$. This together with Result 30 implies that $\prod_{\Lambda_{r-1}} \rho_j"tf"\rho_i$; that is, $\tau_{r-1}"tf"\rho_i$ for every $i \in \Lambda_r$.

3. From 1 we know $m_{tf}(I) \leq \tau_1$. Assume $m_{tf}^{r-1}(I) \leq \tau_{r-1}$ for every r such that $2 \leq r < k$ where k is given by Definition 13. Show $m_{tf}^r(I) \leq \tau_r$. From 2 $m_{tf}(\tau_{r-1}) \leq \prod_{\Lambda_r} \rho_i = \tau_r$. From the inductive hypothesis $\tau_{r-1} \geq m_{tf}^{r-1}(I)$ and therefore, from Result 31 we deduce that $m_{tf}[m_{tf}^{r-1}(I)] \leq \tau_r$ or equivalently $m_{tf}^r(I) \leq \tau_r$.

4. Show $m_{tf}^{q-1}(I) = \emptyset$. From 4 of Definition 13 $\prod_{\Lambda_k} \rho_i = \tau_k = \emptyset$. Hence $m_{tf}^k(I) \leq \tau_k = \emptyset$. Therefore $m_{tf}^k(I) = \emptyset$. From the comments after Result 32 this implies that $m_{tf}^{q-1}(I) = \emptyset$.

Suppose $m_{tf}^{q-1}(I) = \emptyset$. Let k be the first integer such that $m_{tf}^{k}(I) = \emptyset$. Then $k \leq q-1$.

1. Let $E_1 = \{\rho_i | i \epsilon \Lambda_1\}$ be a set of partitions with the properties that $I"tf"\rho_i$ for every $i \epsilon \Lambda_1$ and $\prod_{\Lambda_1} \rho_i = m_{tf}(I)$. Such a set exists since $\{\rho | I"tf"\rho\}$ has the properties. However, it should be noted that one may not need to include all of these partitions in E_1.

2. Let $E_2 = \{\rho_i | i \epsilon \Lambda_2\}$ be a set of partitions with the properties that $E_2 > E_1$, $m_{tf}(I)"tf"\rho_i$ for every $i \epsilon \Lambda_2$ and $\prod_{\Lambda_2} \rho_i = m_{tf}^{2}(I)$. Again such a set exists since $E_2 = \{\rho | m_{tf}(I)"tf"\rho\}$ has the required properties. This follows from the fact that $m_{tf}^{2}(I) < m_{tf}(I)$.

3. Let $E_k = \{\rho_i | i \epsilon \Lambda_k\}$ be a set of partitions with the properties that $E_k > E_{k-1}$, $m_{tf}^{k-1}(I)"tf"\rho_i$ for every $i \epsilon \Lambda_k$ and $\prod_{\Lambda_k} \rho_i = m_{tf}^{k}(I)$. The set $E_k = \{\rho | m_{tf}^{k-1}(I)"tf"\rho\}$ satisfies these properties since $m_{tf}^{k}(I) < m_{tf}^{k-1}(I)$. Again it should be noted that one may not need to include all these partitions in E_k.

4. For every i such that $i \epsilon \Lambda_k$, let $h_i(a) = h_i(b) \Leftrightarrow \rho_i[a] = \rho_i[b]$; i.e. let h_i be the function implies by ρ_i. Since $\prod_{\Lambda_k} \rho_i = m_{tf}^{k}(I) = \emptyset$, h is 1-1. Note that the range of h is $\{0,1\}^{j_k}$. Thus if we let n be the number of elements in Λ_k to be consistent in notation then $h:\{s\} \rightarrow \{0,1\}^{n}$.

5. From Result 28 since $I"tf"\rho_i$ for every $i \epsilon \Lambda_1$ we know that $T_i(y,x) = G_i(f(y,x),x)$ for every $y \epsilon \{0,1\}^{n}$. Also from Result 28 since $\prod_{\Lambda_{r-1}} \rho_i = m_{tf}^{r-1}(I)$ and $m_{tf}^{r-1}(I)"tf"\rho_i$ when $i \epsilon \Lambda_r$ we know that

$T_i(y,x) = G_i(y_{\Lambda_{r-1}}, f(y,x), x)$ for every $i \in \Lambda_r$. Therefore Definition 13 is satisfied and M can be realized with trigger flip-flop using f for feedback. ‖

An example of this result is given in Figure 12 by machine H. $\{s\} = \{1,2,3,4,5\}$ and δ is given in the figure. Also a function $f:\{s\} \times \{x\} \to \{0,1\}$ is given in Figure 12. In machine H I"tf"$(\overline{1,2};\overline{3,4,5})$ and $m_{tf}(I) = (\overline{1,2};\overline{3,4,5})$ since this is the only partition with this property. $(\overline{1,2};\overline{3,4,5})$"tf"$\rho$ iff ρ is one of $(\overline{1,2};\overline{3,4,5})$, $(\overline{1,4};\overline{2,3,5})$ or $(\overline{2,4};\overline{1,3,5})$ and therefore $m_{tf}^2(I) = (\overline{1};\overline{2};\overline{3,5};\overline{4})$. $(\overline{1};\overline{2};\overline{3,5};\overline{4})$"tf"$\rho$ for every two block partition ρ and therefore $m_{tf}^3(I) = \emptyset$ which implies ρ can be realized with trigger flip-flops using f for feedback. Let $\rho_1 = (\overline{1,2};\overline{3,4,5})$, $\rho_2 = (\overline{1,4};\overline{2,3,5})$ and $\rho_3 = (\overline{1,2,3,4};\overline{5})$. Then $\Lambda_1 = \{1\}$, $\Lambda_2 = \{1,2\}$ and $\Lambda_3 = \{1,2,3\}$ satisfies the properties given in the proof of Result 33. In this case $E_1 = \{\rho_1\}$, $E_2 = \{\rho_1, \rho_2\}$ and $E_3 = \{\rho_1, \rho_2, \rho_3\}$. A coding function h corresponding to ρ_1, ρ_2 and ρ_3 is given in Figure 12.

If a machine can be realized using f for feedback and f is a constant, then we say f can be realized without feedback. Machine J in Figure 13 gives a machine which can be realized without feedback using trigger flip-flops when f is any constant function. Note that I"tf"ρ iff $\rho = (\overline{1,2};\overline{3,4,5})$ and $(\overline{1,2};\overline{3,4,5})$"tf"$\rho$ iff $\rho = (\overline{1,3,4};\overline{2,5})$, $(\overline{1,5};\overline{2,3,4})$ or $(\overline{1,2};\overline{3,4,5})$ and finally $(\overline{1};\overline{2};\overline{3,4};\overline{5})$"tf"$\rho$ for every ρ which is a two block partition. Therefore $m_{tf}^1(I) = (\overline{1,2};\overline{3,4,5})$, $m_{tf}^2(I) = (\overline{1};\overline{2};\overline{3,4};\overline{5})$ and $m_{tf}^3(I) = \emptyset$.

States	Inputs 0	1
1	1	3
2	3	4
3	4	1
4	5	2
5	1	2

δ

States	Inputs 0	1
1	0	0
2	1	0
3	0	1
4	0	1
5	1	0

f

Machine H.

$h(1) = (0,0,0)$

$h(2) = (0,1,0)$

$h(3) = (1,1,0)$

$h(4) = (1,0,0)$

$h(5) = (1,1,1)$

$$T_1(y_1,y_2,y_3,x) = \bar{x}\, f(y_1,y_2,y_3,x) + x$$

$$T_2(y_1,y_2,y_3,x) = y_1\, f(y_1,y_2,y_3,x) + x\bar{y}_1$$

$$T_3(y_1,y_2,y_3,x) = y_1 x\, f(y_1,y_2,y_3,x) + y_1\bar{x}\bar{y}_2 + y_1\bar{x}\, f(y_1,y_2,y_3,x)$$

		0	1	0	1	0	1
1	000	000	110	000	110	0	0
2	010	110	100	100	110	1	0
3	110	100	000	010	110	0	1
4	100	111	010	011	110	0	1
5	111	000	010	111	101	1	0
		Y		T		f	

Figure 12.

States	Inputs 0	Inputs 1	
1	5	2	h(1) = (0,0,0)
2	3	1	h(2) = (0,1,0)
3	d	4	h(3) = (1,1,1)
4	1	3	h(4) = (1,1,0)
5	2	5	h(5) = (1,0,0)

Machine J.

	Inputs 0	Inputs 1	Inputs 0	Inputs 1
000	100	010	100	010
010	111	000	101	010
111	d	110	d	001
110	000	111	110	001
100	010	100	110	000
	(Y_1, Y_2, Y_3)		(T_1, T_2, T_3)	

$$T_1(y_1, y_2, y_3, x) = \bar{x}$$

$$T_2(y_1, y_2, y_3, x) = x\bar{y}_1 + \bar{x}y_1$$

$$T_3(y_1, y_2, y_3, x) = y_2\bar{y}_1\bar{x} + xy_1y_2$$

Figure 13.

Now we relate feedback for the unit delay case to feedback with trigger flip-flop memory elements. One result we get is that the set of machines which can be realized without feedback using unit delays and the set of machines which can be realized without feedback

using trigger flip-flops are disjoint when the machines are completely specified. Before we prove this result we consider a lemma.

Result 34 (Lemma).

Let M be a machine, $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$ and τ "tf" ρ. Let $a,b \in \{s\}$ such that $\tau[a] = \tau[b]$. If there exists $x \in \{x\}$ such that $f(a,x) = f(b,x)$ and $\delta(a,x) = \delta(b,x)$ then $m_{tf}(\tau)[a] = m_{tf}(\tau)[b]$.

Proof.

Consider any ρ such that τ "tf" ρ. If $\rho[a] \neq \rho[b]$ then $\rho[\delta(a,x)] \neq \rho[\delta(a,x)]$ which is impossible since $\delta(a,x) = \delta(b,x)$. Thus $\rho[a] = \rho[b]$ which implies $m_{tf}(\tau)[a] = m_{tf}(\tau)[b]$.

If $\delta:\{s\} \times \{x\} \rightarrow \{s\}$ and $\{x_i\}_1^n$ is a sequence in $\{x\}$ we define $\delta(a,\{x_i\}_1^n) = a$ if $n = 0$, $\delta(a,\{x_i\}_1^n) = \delta(a,x_1)$ if $n = 1$ and if $n > 1$ we inductively define $\delta(a,\{x_i\}_1^n) = \delta[\delta(a,\{x_i\}_1^{n-1}),x_n]$. With this notation we can prove the following result.

Result 35 (Lemma).

Let M be a completely specified machine and $f:\{s\}\times\{x\} \rightarrow \{0,1\}^{\ell}$. Let there exist $a,b \in \{s\}$ with $a \neq b$ and a sequence $\{x_i\}_1^{q-1}$ with $q \geq 2$ in $\{x\}$ such that $\delta(a,\{x_i\}_1^j)$ and $\delta(b,\{x_i\}_1^j)$ are specified when $1 \leq j \leq q-1$ and, in addition, $f[\delta(a,\{x_i\}_1^{j-1}),x_j] = f[\delta(b,\{x_i\}_1^{j-1}),x_j]$ for every j such that $1 \leq j \leq q-1$. Then $m_{pf}^{q-1}(I) = \emptyset$ implies that $m_{tf}^{q-1}(I) \neq \emptyset$.

Proof.

Since $I[a] = I[b]$, the hypothesis implies that $m^j_{pf}(I)[a] = m^j_{pf}(I)[b]$ for all $j \leq q-1$. Thus $m^{q-1}_{pf}(I) = \emptyset$ and the hypothesis implies that $\delta(a, \{x_i\}_1^{q-1}) = \delta(b, \{x_i\}_1^{q-1})$. Let r be the first integer such that $\delta(a, \{x_i\}_1^r) = \delta(b, \{x_i\}_1^r)$. Then $1 \leq r \leq q-1$ and if $a_1 = \delta(a, \{x_i\}_1^{r-1})$ and $b_1 = \delta(b, \{x_i\}_1^{r-1})$ then $a_1 \neq b_1$ and $\delta(a_1, x_r) = \delta(b_1, x_r)$. Show $m^j_{tf}(I)[a_1] = m^j_{tf}(I)[b_1]$ for every j such that j is a positive integer. Clearly $I[a_1] = I[b_1]$ and since $f(a_1, x_r) = f(b_1, x_r)$ and $\delta(a_1, x_r) = \delta(b_1, x_r)$ from Result 34 this implies that $m_{tf}(I)[a_1] = m_{tf}(I)[b_1]$. Suppose $m^k_{tf}(I)[a_1] = m^k_{tf}(I)[b_1]$. Again from Result 34 this implies that $m_{tf}(m^k_{tf}(I))[a_1] = m_{tf}(m^k_{tf}(I))[b_1]$ or $m^{k+1}_{tf}(I)[a_1] = m^{k+1}_{tf}(I)[b_1]$. If we let $j = q-1$ then $m^{q-1}_{tf}(I)[a_1] = m^{q-1}_{tf}(I)[b_1]$ where $a_1 \neq b_1$. Therefore $m^{q-1}_{tf}(I) \neq \emptyset$. ‖

From Result 35 we get the following result on feedback free machines.

Result 36 (Corollary).

Let M be a completely specified sequential machine with q states and $q > 1$.

i) If $m^{q-1}_{pf}(I) = \emptyset$, then $m^{q-1}_{tf}(I) \neq \emptyset$.

ii) If $m^{q-1}_{tf}(I) = \emptyset$, then $m^{q-1}_{pf}(I) \neq \emptyset$.

In particular the set of machines which can be realized without feedback using unit delays is disjoint from the set of machines that

can be realized without feedback using trigger flip-flop memory elements.

Proof.

i) Since $q > 1$ there exists $a, b \in \{s\}$ such that $a \neq b$. Let $\{x_i\}_1^{q-1}$ be any sequence in $\{x\}$. Since f is a constant, Result 35 holds and $m_{tf}^{q-1}(I) \neq \emptyset$.

ii) Suppose $m_{pf}^{q-1}(I) = \emptyset$, then from i $m_{tf}^{q-1}(I) \neq \emptyset$ which is a contradiction. Therefore $m_{pf}^{q-1}(I) = \emptyset$.

The last statement follows from Result 33 and 26. ‖

It is clear Result 36 does not necessarily hold if the machine is not completely specified. For example, one could consider a machine where $\delta(a,x)$ is not specified for any state a and input x. We now turn our consideration to the computation of $m_{tf}(\tau)$.

Definition 15. Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Let τ be a state partition.

1. Let $A(\tau) = \{(b,c) | \tau[b] = \tau[c]$ and there exists x such that $f(c,x) = f(b,x)$ and $\delta(b,x) = \delta(c,x)\} \cup \{(b,c) |$ there exists $a \in \{s\}$ and input x such that $f(a,x) = f(b,x)$, $\tau[a] = \tau[b]$ and $c = \delta(a,x)$ while $a = \delta(b,x)$ or $\delta(a,x) = a$ and $\delta(b,x) = c\}$.

2. Let $A^{\#}(\tau)$ be the smallest equivalence class which contains $A(\tau)$ and let β_1 be the state partition implied by $A^{\#}(\tau)$.

Result 37.

If τ abd β_1 are defined as in Definition 15 then $\beta_1 \leq m_{tf}(\tau)$.

Proof.

Let $b,c \in \{s\}$ such that $\beta_1[b] = \beta_1[c]$. Let ρ be a partition such that τ "tf" ρ.

i) Suppose $(b,c) \in A(\tau)$. If $\tau[b] = \tau[c]$ and there exists x such that $f(c,x) = f(b,x)$ and $\delta(c,x) = \delta(b,x)$ then $\rho[b] = \rho[c]$ from Result 34. If there exists $a \in \{s\}$ and $x \in \{x\}$ such that $f(a,x) = f(b,x)$, $\tau[a] = \tau[b]$ and $c = \delta(a,x)$ while $a = \delta(b,x)$, then if $\rho[a] = \rho[b]$ we have that $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ since $\tau \cdot \rho$ "pf" ρ or $\rho[c] = \rho[a] = \rho[b]$. If $\rho[a] \neq \rho[b]$, then $\rho[\delta(a,x)] \neq \rho[\delta(b,x)]$ or $\rho[c] \neq \rho[a]$ which implies $\rho[b] = \rho[c]$ since ρ has only two blocks. Therefore $\rho[b] = \rho[c]$. The proof for the case $a = \delta(b,x)$ and $\delta(b,x) = c$ is identical.

ii) Suppose $b = a_o$, $c = a_{k+1}$ $k \geq 0$ and $(a_o,a_1),(a_1,a_2),\ldots,(a_k,a_{k+1})$ are all in $A(\tau)$. Then from i) $\rho[a_i] = \rho[a_{i+1}]$ for every i $0 \leq i \leq k$. Since ρ is a partition, this implies $\rho[b] = \rho[a_o] = \rho[a_{k+1}] = \rho[c]$.

Cases i, ii cover all cases for b and c such that $\beta_1[b] = \beta_1[c]$ except when b = c which is obvious. Thus $\beta_1[b] = \beta_1[c]$ implies $\rho[b] = \rho[c]$ for every ρ such that τ "pf" ρ. Therefore $m_{tf}(\tau)[b] = m_{tf}(\tau)[c]$. This implies that $\beta_1 \leq m_{tf}(\tau)$. ‖

It should be noted that $A(\tau)$ can be determined by inspection. One has only to observe that $(b,c) \in A(\tau)$ if $\delta(b,x) = \delta(c,x)$ for some x with $f(a,x) = f(b,x)$ and if any two of a, b, $\delta(a,x)$, $\delta(b,x)$ are equal when $\tau[a] = \tau[b]$ and $f(a,x) = f(b,x)$ then the other two are a pair in $A(\tau)$.

Definition 16. Let M be a machine and let τ, β_i be state partitions such that $m_{tf}(\tau) \geq \beta_i$ where $i \geq 1$.

1. Let $B(\beta_i) = \{(\beta_i[b], \beta_i[c]) \mid \tau[b] = \tau[c]$ and there exists input x such that $f(b,x) = f(c,x)$ and $\beta_i[\delta(b,x)] = \beta_i[\delta(c,x)]\} \cup \{\beta_i(b), \beta_i(c)) \mid \beta_i(c) = \beta_i[\delta(a,x)]$ and $\beta_i[a] = \beta_i[\delta(b,x)]$ or $\beta_i[a] = \beta_i[\delta(a,x)]$ and $\beta_i[c] = \beta_i[\delta(b,x)]$ for $a,b \in \{s\}$ and $x \in \{x\}$ such that $\tau[a] = \tau[b]$, $f(a,x) = f(b,x)\}$.

2. Let $B^{\#}(\beta_i)$ be the smallest equivalence relation which contains $B(\beta_i)$. Let β_{i+1} be a partition on $\{s\}$ defined by $\beta_{i+1}[a] = \beta_{i+1}[b]$ iff $(\beta_i[a], \beta_i[b]) \in B^{\#}(\beta_i)$.

Result 38.

$$\beta_{i+1} \leq m_{tf}(\tau) \text{ and } \beta_{i+1} \geq \beta_i.$$

Proof.

Let $b,c \in \{s\}$ such that $\beta_{i+1}[b] = \beta_{i+1}[c]$ which implies that $(\beta_i[c], \beta_i[b]) \in B^{\#}(\beta_i)$. Let ρ be a partition such that τ "pf" ρ.

i) Suppose $(\beta_i[c], \beta_i[b]) \in B(\beta_i)$. If $\tau[b] = \tau[c]$ and there exists x such that $f(b,x) = f(c,x)$ and $\beta_i[\delta(b,x)] = \beta_i[\delta(c,x)]$, then $\rho[\delta(b,x)] = \rho[\delta(c,x)]$ since $\rho \geq \beta_i$. Because τ "tf" ρ, $\rho[b] \neq \rho[c]$ implies

$\rho[\delta(b,x)] \neq \rho[\delta(c,x)]$, which is a contradiction, thus $\rho[b] = \rho[c]$. If there exists a $\epsilon\{s\}$, $x \epsilon\{x\}$ such that $\tau[a] = \tau[b]$, $f(a,x) = f(b,x)$ and $\beta_i[c] = \beta_i[\delta(a,x)]$ while $\beta_i[a] = \beta_i[\delta(b,x)]$; then $\rho[a] = \rho[\delta(b,x)]$ and $\rho[c] = \rho[\delta(a,x)]$ since $\rho \geq \beta_i$. If $\rho[a] = \rho[b]$ then $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ since $\tau \cdot \rho$"pf"ρ which implies $\rho[a] = \rho[\delta(a,x)] = \rho[c]$. Therefore $\rho[b] = \rho[c]$. If $\rho[a] \neq \rho[b]$ then $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ since τ"tf"ρ. This implies $\rho[c] \neq \rho[a]$ and therefore $\rho[b] = \rho[c]$ since ρ has only two blocks. Therefore $(\beta_i[c], \beta_i[b]) \epsilon B(\beta_i)$ implies $\rho[b] = \rho[c]$. The case when $\beta_i[a] = \beta_i[\delta(a,x)]$ and $\beta_i[c] = \beta_i[\delta(b,x)]$ is proved in a similar manner.

ii) Suppose $\beta_i[c] = \beta_i[a_o]$, $\beta_i[b] = \beta_i[a_{k+1}]$ and $(\beta_i[a_o], \beta_i[a_1])$, $(\beta_i[a_1], \beta_i[a_2]) \ldots (\beta_i[a_k], \beta_i[a_{k+1}])$ are such that $(\beta_i[a_j], \beta_i[a_{j+1}] \epsilon B(\beta_i)$ when $0 \leq j \leq k$. Then from i) $\rho[a_j] = \rho[a_{j+1}]$ for every j such that $0 \leq j \leq k$. Therefore $\rho[c] = \rho[a_o] = \rho[a_{k+1}] = \rho[b]$.

Parts i, ii imply that if $(\beta_i[c], \beta_i[b]) \epsilon B^{\#}(\beta_i)$, then $\rho[b] = \rho[c]$ which implies that $m_{tf}(\tau)[b] = m_{tf}(\tau)[c]$. Therefore $m_{tf}(\tau) \geq \beta_{i+1}$.

Results 37 and 38 imply a way to compute $m_{tf}(\tau)$. First compute β_1 as in Definition 15. Then using β_1 compute β_2 as in Definition 16. Continue until $\beta_{i+1} = \beta_i$ for some $i \geq 1$. This must happen since $\beta_{i+1} \geq \beta_i$ for every i and $\{s\}$ is finite. Let $\theta(\tau) = \beta_i$. One must then consider only those $\rho \geq \theta(\tau)$ to see if τ"tf"ρ when one computes $m_{tf}(\tau)$.

For an example of this consider machine H in Figure 12. Here it is seen by inspection that $A(I) = \{(3,4),(4,5),(3,5)\}$ which implies

$A^{\#}(I) = \{(1,1),(2,2),(3,3),(4,4),(5,5),(3,4),(4,5),(3,5),(5,3),(5,4),(4,3)\}$ and $\beta_1 = (\overline{1;2};\overline{3,4,5})$. β_1 could have been easily determined from A(I). Now compute β_2. Again by inspection of Figure 12 $B(\beta_1) = \{(\overline{1;2})\}$ and $B^{\#}(\beta_1) = \{(\overline{1;2}),(\overline{3,4,5;3,4,5}),(\overline{2;1}),(\overline{1;1}),(\overline{2;2})$. Therefore $\beta_2 = (\overline{1,2;3,4,5})$. Compute β_3. By inspection $B(\beta_2) = \varphi$ and therefore $B^{\#}(\beta_2) = \{(\overline{1,2,1,2}),(\overline{3,4,5,3,4,5})\}$. Hence $\beta_3 = (\overline{1,2;3,4,5}) = \beta_2$ and therefore $\theta(I) = \beta_2$. From Results 37, 38 we know that $\theta(I) \leq m_{tf}(I)$. Hence to compute $m_{tf}(I)$, we need only consider all 2 block state partitions ρ such that $\rho > \theta(I)$ and an easy check shows I"tf"$\theta(I)$ and hence $m_{tf}(I) = (\overline{1,2;3,4,5})$. We have made this computation longer than needed. $B(\beta_i)$ can be written down by inspection of the state table and β_{i+1} can be written down directly from $B(\beta_i)$ without looking at $B^{\#}(\beta_i)$.

Let us again consider machine H in Figure 12. We have already determined that when we begin with I $\theta(I) = (\overline{1,2;3,4,5})$. Repeat the calculations this time beginning with $\tau = \theta(I)$. Then by inspection $A(\tau) = \{(3,5)\}$ and $\beta_1 = (\overline{1;2};\overline{3,5;4})$. Continuing $B(\beta_1) = \varphi$ and therefore $\beta_2 = \beta_1$. Thus when we begin with $\tau = \theta(I)$ we get $\theta(\tau) = \theta(\theta(I)) = (\overline{1;2};\overline{3,5;4})$ which we label as $\theta^2(I)$. $\theta^2(I)$ must be less than $m_{tf}{}^2(I)$. Repeat the process letting $\tau = \theta^2(I)$. By inspection of Figure 12 $A(\tau) = \varphi$ and therefore $\beta_1 = \beta_2 = (\overline{1;2};\overline{3;4;5})$. Thus $\theta(\tau) = \theta(\theta^2(I)) = \theta^3(I) = (\overline{1;2};\overline{3;4;5})$. And $\theta^3(I) \leq m_{tf}{}^3(I)$. In this case $\theta^1(I) = m_{tf}(I)$, $\theta^2(I) = m_{tf}{}^2(I)$ and $\theta^3(I) = m_{tf}{}^3(I)$. To check a machine for feedback one should do a computation

as above. If one does not end up with the $\emptyset$ partition, the machine cannot be realized with trigger flip-flops using f for feedback. If one does end up with the $\emptyset$ partition, then he must continue to investigate by, for example, considering those two block partitions $\rho \geq \theta(I)$ to see if $I"tf"\rho$.

Feedback in Set-Reset Flip-Flop Realizations

In order to determine when a function f can be used as feedback in a machine M realized with set-reset memory elements we define the following relations.

Definition 17. Let τ and ρ be state partitions in machine M and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$ where ℓ is a positive integer. Then $\tau"rf"\rho$ iff

1. ρ is a 2 block partition
2. $\tau \cdot \rho"pf"\rho$.
3. For every two states a, b and every input x such that $\delta(a,x)$ and $\delta(b,x)$ are specified, $\tau[a] = \tau[b]$, $\rho[a] \neq \rho[b]$ and $f(a,x) = f(b,x)$; then $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ or $\delta(a,x) \in \rho[a]$ and $\delta(b,x) \in \rho[b]$.

Before we consider the subject of feedback in set-reset realizations we must prove the next results which are similar to the ones proved in the trigger flip-flop development.

Result 39 (Lemma).

Let M be a machine and let $h:\{s\} \rightarrow \{0,1\}^n$ and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Let $\Lambda \leq \{1,\ldots,n\}$ and $g \in \{1,\ldots,n\}$ but $g \notin \Lambda$. If for every $y \in \{0,1\}^n$ and for every input x $Y_g(y,x) = y_g\, U(y_\Lambda, f(y,x),x) + \bar{y}_g\, W(y_\Lambda, f(y,x),x)$ where $U \geq W$, then $R_g(y,x) = I_g(y_\Lambda, f(y,x),x)$ and $S_g(y,x) = H_g(y_\Lambda, f(y,x),x)$.

Proof.

Let $R_g(y,x) = \bar{U}(y_\Lambda, f(y,x),x)$ and $S_g(y,x) = W(y_\Lambda, f(y,x),x)$. Show that this is allowable. That is, show that if $y_g = 1$ and $Y_g(y,x) = 0$ then $R_g(y,x) = 1$ and if $y_g = 0$ and $Y_g(y,x) = 1$ then $S_g(y,x) = 1$. In addition, one must show that R_g and S_g are not both one for any (y,x). Suppose $Y_g(y,x) = 0$. Then if $y_g = 1$ $U(y_\Lambda, f(y,x),x) = 0$. Thus $\bar{U}(y_\Lambda, f(y,x),x) = 1 = R_g(y,x)$. Suppose $Y_g(y,x) = 1$ and $y_g = 0$. Then $W(y_\Lambda, f(y,x)) = 1 = S_g(y,x)$. If $R_g(y,x) = 1$ then $U(y_\Lambda, f(y,x),x) = 0$ and since $U \geq W$ $W(y_\Lambda, f(y,x),x) = 0$ which implies $S_g(y,x) = 0$. If $S_g(y,x) = 1$ then $W(y_\Lambda, f(y,x),x) = 1$ and since $U \geq W$ $U(y_\Lambda, f(y,x),x) = 1$ which implies $R_g(y,x) = 0$. ‖

Result 40 (Theorem).

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Let M be coded by h into $\{0,1\}^n$. Let $\tau = \prod_{\Lambda} \rho_i$ where $\Lambda \leq \{1,\ldots,n\}$ and $\rho = \rho_g$ where $g \in \{1,\ldots,n\}$. If τ "rf" ρ then $R_g(y,x) = I_g(y_\Lambda, f(y,x),x)$ and $S_g(y,x) = H_g(y_\Lambda, f(y,x),x)$. •

Proof.

i) Suppose $g \in \Lambda$. Then $\rho \geq \tau$ and since τ "rf" ρ implies $\tau \cdot \rho$ "pf" ρ we deduce that τ "pf" ρ. Therefore $Y_g(y,x) = F_g(y_\Lambda, f(y,x), x)$ from Result 25. Define $R_g(y,x) = \bar{F}_g(y_\Lambda, f(y,x), x)$ and $S_g(y,x)$ $= F_g(y_\Lambda, f(y,x), x)$. ‖

ii) Suppose $g \notin \Lambda$. Then again $\tau \cdot \rho$ "pf" ρ and from Result 25 $Y_g(y,x) = F_g(y_\Lambda, y_g, f(y,x), x)$. If we let $U(y_\Lambda, d, x) = F_g(y_\Lambda, y_g = 1, d, x)$ for every $d \in \{0,1\}^\ell$ and $W(y_\Lambda, d, x) = F_g(y_\Lambda, y_g = 0, d, x)$ for every $d \in \{0,1\}^\ell$ then $Y_g(y,x) = y_g \; U(y_\Lambda, f(y,x), x) + \bar{y}_g \, W(y_\Lambda, f(y,x), x)$. Recall that there are freedoms on F_g in the proof of Result 25. Namely, $F_g(h_\Lambda(a), h_g(a), f(h(a),x), x) = h_g[\delta(a,x)]$ if $\delta(a,x)$ is specified. For every other (y_Λ, y_g, d, x) with $d \in \{0,1\}^\ell$ and $y_\Lambda \in \{0,1\}$ F_g can be specified in any manner. We specify it as follows:

For every (y_Λ, y_g, d, x) define $F_g(y_\Lambda, y_g, d, x) = F_g(y_\Lambda, \bar{y}_g, d, x)$ when there is no $a \in \{s\}$, $x \in \{x\}$ such that $f(a,x) = d$, $h_\Lambda(a) = y_\Lambda$ and $\delta(a,x)$ is specified. Show when F_g is so specified that $U \geq W$. Suppose $W(y_\Lambda, d, x) = 1$. Then $F_g(y_\Lambda, y_g = 0, d, x) = 1$. Claim $F_g(y_\Lambda, y_g = 1, d, x)$ $= 1$. This clearly is true from the above statements unless there exists $a, b \in \{s\}$ and $x \in \{x\}$ such that $(y_\Lambda, y_g = 0) = (h_\Lambda(a), h_g(a))$, $(y_\Lambda, y_g = 1)$ $= (h_\Lambda(b), h_g(b))$, $f(a,x) = f(b,x) = d$ and $\delta(a,x)$ and $\delta(b,x)$ are specified. But $h_\Lambda(a) = h_\Lambda(b)$ implies $\tau[a] = \tau[b]$ and $h_g(a) = h_g(b)$ implies $\rho[a] \neq \rho[b]$.

Since $h_g(b) = 1$ from $F_g(h_\Lambda(a), h_g(a), f(h(a),x),x) = F_g(y_\Lambda, y_g = 0, d, x) = 1$ we infer that $h_g[\delta(a,x)] = 1$ or $\delta(a,x) \in \rho[b]$. But since τ "rf" ρ this implies $\delta(b,x) \in \rho[b]$ and $h_g[\delta(b,x)] = F_g(h_\Lambda(b), h_g(b), f(h(a),x),x) = F_g(y_\Lambda, y_g = 1, d, x) = 1$. Therefore $W(y_\Lambda, d, x) = 1$ implies that $F_g(y_\Lambda, y_g = 1, d, x) = 1$ which implies $U(y_\Lambda, d, x) = 1$. Therefore $U \geq W$. From Result 39 this implies the theorem. ‖

Result 41 (Theorem).

Let M be a sequential machine coded by h into $\{0,1\}^n$. Let $f: \{s\} \times \{x\} \to \{0,1\}^\ell$. If M is realized by set-reset flip-flops such that $R_g(y,x) = I_g(y_\Lambda, f(y,x),x)$ and $S_g(y,x) = H_g(y_\Lambda, f(y,x),x)$ where $g \in \{1,\ldots,n\}$ and $\Lambda \leq \{1,\ldots,n\}$ then τ "rf" ρ where $\tau = \prod_\Lambda \rho_i$ and $\rho = \rho_g$.

Proof.

i) Suppose $g \in \Lambda$. In general $Y_g(y,x) = y_g \bar{R}_g(y,x) + \bar{y}_g S_g(y,x)$. Therefore $Y_g(y,x) = y_g \bar{I}_g(y_\Lambda, f(y,x),x) + \bar{y}_g H_g(y_\Lambda, f(y,x),x)$ $= F_g(y_\Lambda, f(y,x),x)$.

From Result 25 this implies $\tau \cdot \rho$ "pf" ρ and since $g \in \Lambda$ implies $\rho \geq \tau$ this implies τ "pf" ρ and τ "tf" ρ.

ii) Suppose $g \notin \Lambda$. Then as before $Y_g(y,x) = y_g \bar{I}_g(y_\Lambda, f(y,x)$ $+ \bar{y}_g H_g(y_\Lambda, f(y,x),x) = F_g(y_\Lambda, y_g, f(y,x),x)$. From Result 25 this implies $\tau \cdot \rho$ "pf" ρ. Let $a,b, \in \{s\}$ and $x \in \{x\}$ such that $\tau[a] = \tau[b]$, $\rho[a] \neq \rho[b]$, and $f(a,x) = f(b,x)$. Note that $\tau[a] = \tau[b]$ implies $h_\Lambda(a) = h_\Lambda(b)$ and

$\rho[a] \neq \rho[b]$ implies $h_g[a] \neq h_g[b]$. Suppose $\delta(a,x) \in \rho[b]$ and also suppose $h_g[a] = 1$. Then $h_g[\delta(a,x)] = 0$ which implies $Y_g(h(a),x) = 0 = \bar{I}_g(h_\Lambda(a), f(h(a),x),x)$ or, equivalently, $I_g(h_\Lambda(a), f(h(a),x),x) = 1$. Since $I_g = R_g$ and $R_g = 1$ implies $S_g = 0$ we must have $S_g = H_g(h_\Lambda(a), f(h(a),x),x) = 0$ which in turn equals $H_g(h_\Lambda(b), f(h(b),x),x)$. Therefore $Y_g(h(b),x) = 0$ which implies that $\delta(b,x) \in \rho[b]$. Suppose $h_g[a] = 0$. Then $h_g[\delta(a,x)] = 1$ which implies $Y_g(h(a),x) = 1 = H_g(h_\Lambda(a), f(h(a),x),x)$. This implies that $S_g(h(a),x) = 1$ and therefore $R_g(h(a),x) = 0 = I_g(h_\Lambda(a), f(h(a),x),x)$. Since $I_g(h_\Lambda(a), f(h(a),x) = I_g(h_\Lambda(b), f(h(b),x),x)$ and $h_g(b) = 1$ this implies that $Y_g(h(b),x) = 1$. Therefore $h_g[\delta(b,x)] = 1$ and $\delta(b,x) \in \rho[b]$. Hence τ"rf"ρ.

With these results out of the way we are ready to define the concept of feedback in machines realized with set-reset memory elements. This definition is similar to Definition 13.

Definition 18. Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^\ell$. M can be realized with set-reset flip-flops using f for feedback iff M can be coded by h into $\{0,1\}^n$ such that

1. There exists $\{\Lambda_1,\ldots,\Lambda_k\}$ a set of positive integers such that $u < v$ implies $\Lambda_u < \Lambda_v$.

2. If $i \in \Lambda_1$ then $R_i(y,x) = I_i(f(y,x),x)$ and $S_i(y,x) = H_i(f(y,x),x)$ for every $y \in \{0,1\}^n$ and $x \in \{x\}$.

3. If $i \in \Lambda_r - \Lambda_{r-1}$ where $1 < r \leq k$ then $R_i(y,x) = I_i(y_{\Lambda_{r-1}}, f(y,x),x)$ and $S_i(y,x) = H_i(y_{\Lambda_{r-1}}, f(y,x),x)$.

4. $\prod_{\Lambda_k} \rho_i = \emptyset$ where ρ_i is the partition associated with h_i.

Again we define the m operator this time with respect to "rf", and then prove some results concerning it.

Definition 19. Let M be a machine and $f:\{s\} \times \{x\} \to \{0,1\}^{\ell}$. Let τ be a state partition.

i) $m_{rf}^{1}(\tau) = \prod\{\rho \mid \tau \text{ "rf" } \rho\}$. If $\{\rho \mid \tau \text{ "rf" } \rho\} = \varphi$, then define $m_{rf}^{1}(\tau) = I$. We frequently call $m_{rf}^{1}(\tau)$ by $m_{rf}(\tau)$ deleting the 1.

ii) $m_{rf}^{i+1}(\tau) = m_{rf}(m_{rf}^{i}(\tau))$ for i in $\{1,2,\ldots\}$.

Result 42.

Let τ, τ_1, and ρ be state partitions in a machine M and $f:\{s\} \times \{x\} \to \{0,1\}^{\ell}$. If $\tau_1 \leq \tau$ and τ "rf" ρ, then τ_1 "rf" ρ.

Proof.

Let $a,b \in \{s\}$ such that $\tau_1[a] = \tau_1[b]$, $f(a,x) = f(b,x)$, and $\delta(a,x)$ and $\delta(b,x)$ are specified. $\tau_1[a] = \tau_1[b]$ implies $\tau[a] = \tau[b]$ since $\tau \geq \tau_1$. Suppose $\rho[a] = \rho[b]$, then $\rho[\delta(a,x)] = \rho[\delta(b,x)]$ since $\tau_1 \cdot \rho$ "pf" ρ. Suppose $\rho[a] \neq \rho[b]$ then $\delta(a,x) \in \rho[b]$ implies $\delta(b,x) \in \rho[b]$ since τ "rf" ρ. ‖

Result 43.

Let τ and τ_1 be state partitions. If $\tau \geq \tau_1$ then $m_{rf}(\tau) \geq m_{rf}(\tau_1)$.

Proof.

Let ρ be such that τ "rf" ρ. Then τ_1 "rf" ρ from Result 42 which implies $m_{rf}(\tau) \geq m_{rf}(\tau_1)$. ‖

Result 44.

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Then $m_{rf}^{i+1}(I) \leq m_{rf}^{i}(I)$ for every $i \in \{1,2,\ldots\}$.

Proof.

i) Show $m_{rf}^{2}(I) \leq m_{rf}(I)$. Clearly $m_{rf}(I) \leq I$. Therefore $m_{rf}(m_{rf}(I)) \leq m_{rf}(I)$ from Result 43. Thus $m_{rf}^{2}(I) \leq m_{rf}(I)$.

ii) Suppose $m_{rf}^{j}(I) \leq m_{rf}^{j-1}(I)$ with $2 \leq j$. Then $m_{rf}(m_{rf}^{j}(I)) \leq m_{rf}(m_{rf}^{j-1}(I))$ from Result 43. Thus $m_{rf}^{j+1}(I) \leq m_{rf}^{j}(I)$. ‖

If M is a q state machine then since I can be refined at most q-1 times $m_{rf}^{q-1}(I) = m_{rf}^{q}(I)$. With this we are ready to state and prove the main result of this section.

Result 45 (Theorem).

Let M be a q state machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. M can be realized using set-reset flip-flop memory elements for feedback iff $m_{rf}^{q-1}(I) = \emptyset$.

Proof.

Suppose M can be realized using f for feedback. Then M can be coded by h into $\{0,1\}^{n}$ such that Definition 18 is satisfied. Let

$\tau_r = \prod_{\Lambda_r} \rho_j$.

1. From 2 of Definition 18 and Result 41 I"rf"ρ_i when $i \epsilon \Lambda_1$ and ρ_i is the partition associated with h_i. Therefore, $m_{rf}(I) \leq \prod_{\Lambda} \rho_i = \tau_1$.

2. From 3 of Definition 18 and Result 41 $(\prod_{\Lambda_{r-1}} \rho_j)$"rf"$\rho_i$ for every $i \in \Lambda_r - \Lambda_{r-1}$. This together with Result 42 implies that $\prod_{\Lambda_{r-1}} \rho_j$"tf"$\rho_i$; that is, τ_{r-1}"tf"ρ_i for every $i \epsilon \Lambda_r$.

3. From 1 we know $m_{rf}(I) \leq \tau_1$. Assume $m_{rf}^{r-1}(I) \leq \tau_{r-1}$ for every r such that $2 \leq r < k$ where k is given by Definition 18. Show $m_{rf}^{r}(I) < \tau_r$. From 2 $m_{rf}(\tau_{r-1}) \leq \prod_{\Lambda_r} \rho_i = \tau_r$. From the inductive hypothesis and Result 43 $m_{rf}[m_{rf}^{r-1}(I)] \leq m_{rf}(\tau_{r-1}) \leq \tau_r$. Thus $m_{rf}^{r}(I) \leq \tau_r$.

4. Show $m_{rf}^{q-1}(I) = \emptyset$. From 4 of Definition 18 $\prod_{\Lambda_k} \rho_i = \emptyset$. From 3 $m_{rf}^{k}(I) \leq \tau_k = \prod_{\Lambda_k} \rho_i = \emptyset$. Therefore $m_{rf}^{k}(I) = \emptyset$. From Result 44 this implies $m_{rf}^{q-1}(I) = \emptyset$.

We now consider the converse. Suppose $m_{rf}^{q-1}(I) = \emptyset$. Let k be the first integer such that $m_{rf}^{k}(I) = \emptyset$. Then $1 \leq k \leq q-1$.

1. Let $E_1 = \{\rho_i | i \epsilon \Lambda_1\}$ be a set of partitions with the properties that I"rf"ρ_i for every $i \epsilon \Lambda_1$ and $\prod_{\Lambda_1} \rho_i = m_{rf}(I)$. Such a set exists since the set $\{\rho | I$"rf"$\rho\}$ has the properties.

2. Let $E_2 = \{\rho_i | i \epsilon \Lambda_2\}$ be a set of partitions with the properties that $E_2 > E_1$, $m_{rf}(I)$"rf"ρ_i for every $i \epsilon \Lambda_2$ and $\prod_{\Lambda_2} \rho_i = m_{rf}^{2}(I)$. Again, such a set exists since $\{\rho | m_{rf}(I)$"rf"$\rho\}$ has the desired properties.

This follows from Result 42 and the fact that $m_{rf}^{2}(I) < m_{rf}(I)$.

3. Let $E_k = \{\rho_i \mid i \epsilon \Lambda_k\}$ be a set of partitions with the properties that $E_k > E_{k-1}$, $m_{rf}^{k-1}(I)$"tf"ρ_i for every $i \epsilon \Lambda_k$ and $\prod_{\Lambda_k} \rho_i = m_{rf}^{k}(I)$. The set $\{\rho \mid m_{rf}^{k-1}(I)$"rf"$\rho\}$ has these properties. This follows from Result 42 and the fact that $m_{rf}^{k}(I) < m_{rf}^{k-1}(I)$.

4. For every $i \epsilon \Lambda_k$ let h_i be the function associated with ρ_i, i.e. $h_i:\{s\} \rightarrow \{0,1\}$ and $h_i(a) = h_i(b) \Leftrightarrow \rho_i[a] = \rho_i[b]$. Since $m_{rf}^{k}(I) = \emptyset = \prod_{\Lambda_k} \rho_i$, h is a 1-1 coding function. Note that the range of h is $\{0,1\}^{\Lambda k}$. Thus if we let n be the number of elements in Λ_k to be consistent in notation, then $h:\{s\} \rightarrow \{0,1\}^{n}$.

5. From Result 40 since I"rf"ρ_i for every i in Λ_1 $R_i(y,x) = I_i(f(y,x),x)$ and $S_i(y,x) = H_i(f(y,x),x)$ for every $y \epsilon \{0,1\}^{n}$. Also, from Result 40 since $\prod_{\Lambda_{r-1}} \rho_i = m_{rf}^{r-1}(I)$ and $m_{rf}^{r-1}(I)$"rf"ρ_i, when $i \epsilon \Lambda_r - \Lambda_{r-1}$ and $2 \leq r \leq k$, then $R_i(y,x) = I_i(y_{\Lambda_{r-1}}, f(y,x),x)$ and $S_i(y,x) = H_i(y_{\Lambda_{r-1}}, f(y,x),x)$. Thus Definition 18 is satisfied and M can be realized with set-reset flip-flops using f for feedback. ‖

For an example of the previous results consider machine L in Figure 14. This machine can be realized without feedback using set-reset flip-flops. This can be seen as follows. Let f be a constant. The only partitions ρ such that I"rf"ρ are $\overline{(1,2,3,5;4)}$ and $\overline{(1,2;3,4,5)}$.

Thus $m_{rf}(I) = (\overline{1,2;3,5;4})$. If ρ is one of $(\overline{1,2,3,5;4}), (\overline{1,2;3,4,5})$, $(\overline{1,2,3,4;5}), (\overline{1,4,5:2,3})$ then $(\overline{1,2;3,5;4})$"rf"ρ. Thus $m_{rf}^{2}(I) = \emptyset$ and machine L can be realized without feedback. This is done in Figure 14. Let $\rho_1 = (\overline{1,2;3,4,5})$, $\rho_2 = (\overline{1,2,3,5;4})$ and $\rho_3 = (\overline{1,4,5;2,3})$. Then $\Lambda_1 = \{1,2\}$ and $\Lambda_2 = \{3\}$ satisfies the properties given in the proof of Result 45. In this case $E_1 = \{\rho_1, \rho_2\}$ and $E_2 = \{\rho_3\}$. A coding function h corresponding to ρ_1, ρ_2 and ρ_3 is given in Figure 14.

The following results relate set-reset feedback realizations to unit delay feedback realizations.

Result 46.

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Let τ and γ be state partitions such that τ"pf"γ. If ρ is a two block state partition such that $\rho \geq \gamma$ then τ"rf"ρ.

Proof.

Let $a, b, \in \{s\}$ and let $x \in \{x\}$ such that $\tau[a] = \tau[b]$, $f(a,x) = f(b,x)$ and $\delta(a,x)$ and $\delta(b,x)$ are specified. Since τ"pf"γ, this implies $\gamma[\delta(a,x)] = \gamma[\delta(b,x)]$ and since $\rho \geq \gamma$ this means $\rho[\delta(a,x)] = \rho[\delta(b,x)]$. Therefore τ"rf"γ. ‖

Observe that any state partition $\gamma = \prod\{\rho \mid \rho \geq \gamma$ and ρ is a two block partition$\}$. With this in mind we can easily prove the next result.

Inputs

	0	1
1	5	1
2	5	2
3	3	2
4	4	2
5	3	1

δ

f = constant

$m_{rf}^{1}(I) = \overline{(\overline{1,2};\overline{3,5};\overline{4})}$

$m_{rf}^{2}(I) = (\overline{1};\overline{2};\overline{3};\overline{4};\overline{5})$.

Machine L

h(1) = (0,0,0)

h(2) = (0,0,1)

h(3) = (1,0,1)

h(4) = (1,1,0)

h(5) = (1,0,0)

$R_1 = x \quad S_1 = \bar{x}$

$R_2 = x \quad S_2 = 0$

$R_3 = \bar{x}\bar{y}_1\bar{y}_2$

		0	1	0	1	0	1
1	000	100	000	0dd	ddd	100	000
2	001	100	001	0d1	dd0	100	00d
3	101	101	001	0d0	1d0	d0d	00d
4	110	110	001	00d	110	dd0	001
5	100	101	000	0d0	1dd	dd1	00d
		(Y_1, Y_2, Y_3)		(R_1, R_2, R_3)		(S_1, S_2, S_3)	

Figure 14.

Result 47.

Let M be a machine and $f:\{s\} \times \{x\} \rightarrow \{0,1\}^{\ell}$. Let τ be a state partition. Then $m_{pf}(\tau) \geq m_{rf}(\tau)$.

Proof.

Let γ be a partition such that τ "pf" γ. Let ρ be a 2 block partition greater than γ. Then τ "rf" ρ from Result 46. This implies $m_{pf}(\tau) \geq m_{rf}(\tau)$. ‖

This implies immediately a result relating set-reset flip-flop realizations to unit delay realizations. Results 43 and 47 imply immediately that for every i in $\{1,2,\ldots\}$ that $m^{i}_{pf}(\tau) \geq m^{i}_{rf}(\tau)$. Since $m_{pf}(\tau) \geq m_{rf}(\tau)$ from Result 47 and by induction if $m^{i-1}_{pf}(\tau) \geq m^{i-1}_{rf}(\tau)$ then $m_{pf}[m^{i-1}_{pf}(\tau)] \geq m_{rf}[m^{i-1}_{pf}(\tau)] \geq m_{rf}(m^{i-1}_{rf}(\tau))$ from Results 43 and 47. This implies the following result.

Result 48 (Theorem).

If machine M can be realized with unit delays using f for feedback, then M can be realized with set-reset flip-flop using f for feedback.

Proof.

The hypothesis implies from Result 27 that $m^{q-1}_{pf}(I) = \emptyset$ where q is the number of states. But this implies $m^{q-1}_{rf}(I) = \emptyset$. From Result 45 this implies the theorem. ‖

It should be noticed in Figure 14 that machine L cannot be realized with unit delays using f = constant for feedback. To determine if a machine M can be realized using f for feedback it is necessary to compute $m_{rf}(\tau)$ for various τ which is not an easy problem. In general one must consider all two block partitions ρ to see if τ "rf" ρ. For a q state machine there are $2^{q-1}-1$ such partitions. It is wise to compute $m_{pf}(\tau)$ first. From Result 47 we know that $m_{rf}(\tau) \leq m_{pf}(\tau)$, therefore, it is not necessary to consider these ρ such that $\rho \geq m_{pf}(\tau)$.

In this chapter we have developed a method for determining when a machine can be realized using a function f for feedback with either set-reset or trigger flip-flop memory elements. This method is more difficult to apply than the one given in Reference 3 for the unit delay case. We have also shown in Results 36 and 48 that for a given machine its feedback properties will be different for trigger, set-reset and unit delay type realizations. Thus in general one must first decide the type of memory element he wants to use before making a study of the feedback characteristics of a machine.

BIBLIOGRAPHY

[1] Ginzburg, A., and Yoeli, M., "Products of Automata and the Problem of Covering," Technion, Report No. 15 (July 1963).

[2] Hartmanis, J., "On the State Assignment Problem for Sequential Machines, I.," IRE Trans on Electronic Computers, EC-10, pp. 157-165 (1961).

[3] Hartmanis, J., and Stearns, R. E., Algebraic Structure Theory of Sequential Machines, Prentice Hall, 1966.

[4] Liu, C. L., "Some Memory Aspects of Finite Automata," MIT Research Lab. of Electronics, Report 411 (May 1963).

[5] Stearns, R. E., and Hartmanis, J., "On the State Assignment Problem for Sequential Machines, II.," IRE Trans on Electronic Computers, EC-10, pp. 593-603 (December 1961).

[6] Yoeli, M., "The Cascade Decomposition of Sequential Machines," IRE Trans on Electronic Computers, EC-10, pp. 587-592 (December 1961).

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R&D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
The University of Texas Laboratories for Electronics & Related Science Research Austin, Texas 78712	UNCLASSIFIED 2b. GROUP

3. REPORT TITLE

ON THE STRUCTURE OF SEQUENTIAL MACHINE REALIZATIONS

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Scientific Interim

5. AUTHOR(S) (Last name, first name, initial)

Charles A. Harlow and Clarence L. Coates

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
July 19, 1967	78	6

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
AF-AFOSR-766-67 b. PROJECT NO. 4751 c. 61445014 d. 681305	JSEP, Technical Report No. 27 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) AFOSR-67 1761

10. AVAILABILITY/LIMITATION NOTICES

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
TECH, OTHER	JSEP through AF Office of Scientific Research (SREE) 1400 Wilson Boulevard Arlington, Virginia 22209

13. ABSTRACT

In most studies of the structure of sequential machines there has been a tacit assumption that the machine was to be realized with unit delay memory elements. In this report we consider sequential machines that are realized with either trigger or set-reset flip-flop memory elements.

It is shown that the relation called a partition pair which predicts the dependence of the input functions to unit delay memory elements does not predict the dependence of the input functions to trigger or set-reset flip-flop memory elements. In this paper we define relations called t-pairs and r-pairs which characterize the dependence of the input functions to trigger and set-reset flip-flop memory elements respectively. It is found that these relations do not have all the algebraic properties that partition pairs possess.

Feedback in sequential machines that are realized with trigger or set-reset flip-flop memory elements is also studied. A method is given for determining when a machine can be realized with either trigger or set-reset flip-flop memory elements using function f for feedback. It is shown that if a sequential machine can be realized with unit delay memory elements using a function f for feedback then it can be realized with set-reset flip-flops using f for feedback. It is also shown that for completely specified machines that if a machine can be realized without feedback using unit delay memory elements then it cannot be realized without feedback using trigger flip-flop memory elements. The converse statement is also true.

DD FORM 1 JAN 64 1473

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
SEQUENTIAL MACHINES STRUCTURE THEORY TRIGGER AND SET-RESET FLIP-FLOPS FEEDBACK						

INSTRUCTIONS

1. ORIGINATING ACTIVITY: Enter the name and address of the contractor, subcontractor, grantee, Department of Defense activity or other organization *(corporate author)* issuing the report.

2a. REPORT SECURITY CLASSIFICATION: Enter the overall security classification of the report. Indicate whether "Restricted Data" is included. Marking is to be in accordance with appropriate security regulations.

2b. GROUP: Automatic downgrading is specified in DoD Directive 5200.10 and Armed Forces Industrial Manual. Enter the group number. Also, when applicable, show that optional markings have been used for Group 3 and Group 4 as authorized.

3. REPORT TITLE: Enter the complete report title in all capital letters. Titles in all cases should be unclassified. If a meaningful title cannot be selected without classification, show title classification in all capitals in parenthesis immediately following the title.

4. DESCRIPTIVE NOTES: If appropriate, enter the type of report, e.g., interim, progress, summary, annual, or final. Give the inclusive dates when a specific reporting period is covered.

5. AUTHOR(S): Enter the name(s) of author(s) as shown on or in the report. Enter last name, first name, middle initial. If military, show rank and branch of service. The name of the principal author is an absolute minimum requirement.

6. REPORT DATE: Enter the date of the report as day, month, year; or month, year. If more than one date appears on the report, use date of publication.

7a. TOTAL NUMBER OF PAGES: The total page count should follow normal pagination procedures, i.e., enter the number of pages containing information.

7b. NUMBER OF REFERENCES: Enter the total number of references cited in the report.

8a. CONTRACT OR GRANT NUMBER: If appropriate, enter the applicable number of the contract or grant under which the report was written.

8b, 8c, & 8d. PROJECT NUMBER: Enter the appropriate military department identification, such as project number, subproject number, system numbers, task number, etc.

9a. ORIGINATOR'S REPORT NUMBER(S): Enter the official report number by which the document will be identified and controlled by the originating activity. This number must be unique to this report.

9b. OTHER REPORT NUMBER(S): If the report has been assigned any other report numbers *(either by the originator or by the sponsor)*, also enter this number(s).

10. AVAILABILITY/LIMITATION NOTICES: Enter any limitations on further dissemination of the report, other than those imposed by security classification, using standard statements such as:

(1) "Qualified requesters may obtain copies of this report from DDC."

(2) "Foreign announcement and dissemination of this report by DDC is not authorized."

(3) "U. S. Government agencies may obtain copies of this report directly from DDC. Other qualified DDC users shall request through _____."

(4) "U. S. military agencies may obtain copies of this report directly from DDC. Other qualified users shall request through _____."

(5) "All distribution of this report is controlled. Qualified DDC users shall request through _____."

If the report has been furnished to the Office of Technical Services, Department of Commerce, for sale to the public, indicate this fact and enter the price, if known.

11. SUPPLEMENTARY NOTES: Use for additional explanatory notes.

12. SPONSORING MILITARY ACTIVITY: Enter the name of the departmental project office or laboratory sponsoring *(paying for)* the research and development. Include address.

13. ABSTRACT: Enter an abstract giving a brief and factual summary of the document indicative of the report, even though it may also appear elsewhere in the body of the technical report. If additional space is required, a continuation sheet shall be attached.

It is highly desirable that the abstract of classified reports be unclassified. Each paragraph of the abstract shall end with an indication of the military security classification of the information in the paragraph, represented as *(TS)*, *(S)*, *(C)*, or *(U)*.

There is no limitation on the length of the abstract. However, the suggested length is from 150 to 225 words.

14. KEY WORDS: Key words are technically meaningful terms or short phrases that characterize a report and may be used as index entries for cataloging the report. Key words must be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location, may be used as key words but will be followed by an indication of technical context. The assignment of links, rules, and weights is optional.

GPO 886-551